

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки
Кафедра Автоматики і управління у технічних системах

«На правах рукопису»

УДК _____

«Допущений до захисту»

Завідувач кафедри

_____ Ролік О.І. _____

(підпис) (ініціали, прізвище)

«_____» _____ 2019 р.

Магістерська дисертація

зі спеціальності (спеціалізації) _____ 126 Інформаційні системи та технології _____
(код і назва спеціальності)

на тему: Система управління контентом публічних заходів

Виконав студент II курсу, групи ІА-82мп _____
(шифр групи)

_____ Ілуща Євген Олександрович _____

(прізвище, ім'я, по-батькові)

_____ (підпис)

Керівник _____ к.т.н., доцент. АУТС Дорогий Я.Ю. _____

(посада, наукова ступінь, звання, прізвище, ініціали)

_____ (підпис)

Консультант _____

(назва розділу)

(посада, наукова ступінь, звання, прізвище, ініціали)

_____ (підпис)

Рецензент _____

(посада, наукова ступінь, вчене звання, прізвище, ініціали)

_____ (підпис)

Засвідчую, що у цьому дипломному
проекті немає запозичень з праць
інших авторів без відповідних посилань.

Студент _____
(підпис)

Київ – 2019 р.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

ФІОТ

Кафедра АУТС

Ступінь вищої освіти «магістр»

Зі спеціальності 126 «Інформаційні системи та технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Ролік А.І.

(підпис)

(ініціали, прізвище)

«__» _____.

ЗАВДАННЯ

на магістерську дисертацію студенту

Ілущі Євген Олександровичу

(прізвище, ім'я, по батькові)

1. Тема роботи: Система управління контентом публічних заходів

керівник роботи к.т.н., доцент. АУТС Дорогий Ярослав Юрійович,

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «__» _____ р. №

2. Строк подання студентом проекту _____

3. Вихідні дані до магістерської дисертації _____

4. Перелік завдань, які потрібно розробити: огляд, аналіз основних способів збереження та архівування даних, виявлення недоліків існуючих рішень, розробка інформаційного забезпечення та опис функціональних вимог системи, розробка функціональної схеми, опис реалізації веб-застосунку, її архітектури та інструкція користувача, розробка стартап-преоєкту.

5. Перелік графічного матеріалу функціональна та структурна схема системи, структурна схема компонентів, діаграма послідовності, Use-case діаграма, діаграма класів, діаграма бази даних та схема структурної діяльності

6. Консультанти розділів проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 28.10.2019

Календарний план

№ з/п	Назва етапів виконання дипломного проекту	Строк виконання етапів проекту	Примітка
	Огляд та аналіз основних способів збереження та архівування даних	28.10.2019	
	Розробка інформаційного забезпечення та опис функціональних вимог системи	06.11.2019	
	Опис програмної реалізації та технічного забезпечення	14.11.2019	
	Опис інструкції користувача	22.11.2019	
	Проведення аналізу стартап-проекту	25.11.2019	
	Оформлення ПЗ	03.12.2019	

Студент

Керівник проекту

(підпис)

(підпис)

Ілуца Є.О.

(ініціали, прізвище)

Дорогий Я.Ю.

(ініціали, прізвище)

РЕФЕРАТ

Структура та обсяг роботи. Пояснювальна записка магістерської дисертації освітньо-кваліфікаційного рівня “магістр” на тему: «Система управління контентом публічних заходів», що складається з п’яти розділів, містить 73 сторінки, 10 рисунків, 8 додатків, 8 джерел.

Магістерська дисертація містить результати розроблення програмного продукту, що може бути використаний для збереження публічних заходів.

Метою розробки є створення зручного веб-застосування для організаторів заходів або звичайних користувачів, що мають бажання зберегти захід у вигляді проекту на сайті.

Задачі розробки: дослідження існуючих аналогів, виділення їх основних недоліків та проектування нових функціональних вимог до обраної предметної області; надати користувачам можливість миттєвого перегляду медіафайлів під час детального розглядання заходу; надати користувачам можливість додавання, редагування та видалення статей.

Об’єктом дослідження виступає платформа для збереження публічного заходу.

Предметом дослідження виступають існуючі методи архівування організаційних заходів

ЗБЕРЕЖЕННЯ ПРОЕКТІВ, АРХІВУВАННЯ ПРОЕКТІВ, ПУБЛІЧНІ ЗАХОДИ, ХМАРНЕ СХОВИЩЕ.

ABSTRACT

Structure and scope of work. Explanatory note of the master's dissertation on the topic: "Public events content management system", consists of five sections, contains 73 pages, 10 drawings, 8 appendices, 8 sources.

The master's dissertation contains the results of the development of a software product that can be used to upload public events.

The purpose of the development is to create a convenient web application for event organizers or regular users who want to save the event as a project on the site.

Development tasks: research of existing analogues, highlighting their main disadvantages and designing new functional requirements for the selected subject area; enable users to instantly view media files as they view the event in detail; Enable users to add, edit, and delete articles.

The object of the study is a platform for preserving public events.

The subject of the study is the existing methods of archiving organizational activities

PROJECT UPLOADING, PROJECT ARCHIVING, PUBLIC EVENTS, CLOUD STORAGE.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	8
ВСТУП	9
1 ОГЛЯД ТА АНАЛІЗ ОСНОВНИХ СИСТЕМ збереження даних.....	11
1.1 Огляд хмарних сховищ	11
2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ ТА ОПИС	
ФУНКЦІОНАЛЬНИХ ВИМОГ СИСТЕМИ	24
2.2 Вхідні дані	27
2.3 Вихідні дані.....	28
2.4 Опис структури бази даних.....	28
3 Опис програмної реалізації та технічного забезпечення	31
3.1 Засоби розробки.....	31
3.2 Вимоги до технічного забезпечення	45
3.3 Архітектура програмного забезпечення.....	46
3.3.1 Діаграма класів.....	46
3.3.2 Діаграма компонентів.....	48
3.3.3 Специфікація функцій.....	48
4 Інструкція Користувача Та тестування системи	51
4.1 Use-case діаграма	Ошибкa! Закладка не определена.
4.2.3 Сторінка редагування проекту	53
4.2.4 Сторінка видалення проекту.....	Ошибкa! Закладка не определена.
4.2.6 Сторінка детального перегляду проекту	54
4.2.7 Сторінка детального перегляду проекту	55
4.3 Випробовування програмного продукту.....	56
4.3.1 Мета випробувань.....	56
4.3.2 Загальні положення	56
4.3.3 Результати випробувань.....	56
5 РОЗРОБКА СТАРТАП-ПРОЕКТУ	71
5.1 Вступ	71

5.2	Опис ідеї стартап-проекту	71
5.3	Розроблення маркетингової програми стартап-проекту	87

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ІТ – інформаційні технології

ПЗ – програмне забезпечення

API – Application Programming Interface

IP – Internet Protocol PNI – Private Network Interconnect

ОС – Операційна Система

DB – Database

VM – Virtual Machine

URL – Uniform Resource Locator

SQL – Structured Query Language

DHCP – Dynamic Host Configuration Protocol

UML – Unified Modeling Language

REST – Representational State Transfer

DML – Data Manipulation Language

DDL – Data Definition Language

ORM – Object-Relation Mapping

ВСТУП

Предметом дослідження даної магістерської дисертації є існуючі рішення в задачах управління публічними заходами. Об'єктом дослідження є платформа для управління публічним заходом. Актуальність таких систем є високою, через те, що популярність публічних заходів завжди була у високому попиті, адже прикладів таких подій можна навести дуже багато:

- прес-конференції;
- церемонії відкриття;
- живі виступи;
- церемонії нагородження;
- благодійні акції;
- дитячі свята;
- загальноміські події;
- події зі збору коштів;
- тощо.

У веб-додатках типу «хмарні сховища», ми маємо змогу зберігати інформацію у вигляді файлів, що в більшості випадків є найбільш зручним варіантом. Але якщо мова йде про зберігання проектів одного й того самого формату, виникає необхідність у додаткових функціональних вимогах та прибирання зайвих. Таку проблему, можна спостерігати при архівуванні публічних заходів. Доцільно мати систему, завдяки якій можливо було б мати функціонал, що тісно пов'язаний саме із даною тематикою задачі, а також вдосконалений графічний інтерфейс під проекти такого типу.

Відштовхуючись від вищезазначеного, можемо стверджувати, що існуючі рішення потребують доопрацювання. Саме тому задачею цієї магістерської дисертації є створення суто нового веб-додатку, що покривав би усі функціональні вимоги користувачів під час вирішення задач управління контентом публічних заходів.

Створена система матиме практичну цінність у вигляді допомоги організаторам та компаніям, що спеціалізуються на створенні заходів, зручного збереження та наукову цінність користувачам, що розглядатимуть проекти в системі заради власного розвитку у побудові власних нових заходів, опираючись на наробітки інших користувачів.

1 ОГЛЯД ТА АНАЛІЗ ОСНОВНИХ СИСТЕМ ЗБЕРЕЖЕННЯ ДАНИХ

1.1 Огляд хмарних сховищ

Хмарне сховище даних - модель онлайн-сховища, в якому дані зберігаються на численних розподілених в мережі серверах, що надаються в користування клієнтам, в основному, третьою стороною. На противагу моделі зберігання даних на власних виділених серверах, придбаних або орендованих спеціально для подібних цілей. Дані зберігаються, а так само і обробляються, в так званому хмарі, яке представляє собою, з точки зору клієнта, один великий віртуальний сервер. Фізично ж такі сервери можуть розташовуватися віддалено один від одного географічно, аж до розташування на різних континентах. Процес завантаження файлів приблизно такий самий, як і в соціальних мережах. Їх відмінність полягає в більшій надійності хмарний середовищ та в більшій зручності управління даними в соціальних мережах.

1.1.1 Google Drive

Google Drive - це послуга зберігання та синхронізації файлів, розроблена Google. Google Drive дозволяє користувачам зберігати файли на своїх серверах, синхронізувати файли на різних пристроях та обмінюватися файлами. Окрім веб-сайту, Google Drive пропонує додатки з офлайн-можливостями для комп'ютерів Windows та macOS, а також смартфонів та планшетів Android та iOS. Google Drive включає в себе Документи Google, Таблиці Google і Google Слайди, які є частиною офісного набору, що дозволяє спільне редагування документів, електронних таблиць, презентацій, малюнків, форм та іншого.

Google Drive включає систему спільного використання файлів, у якій автор файлу чи папки за замовчуванням є його власником.[1]

Загальна схема спільного використання файлів виглядає наступним чином (рисунок 1.1):

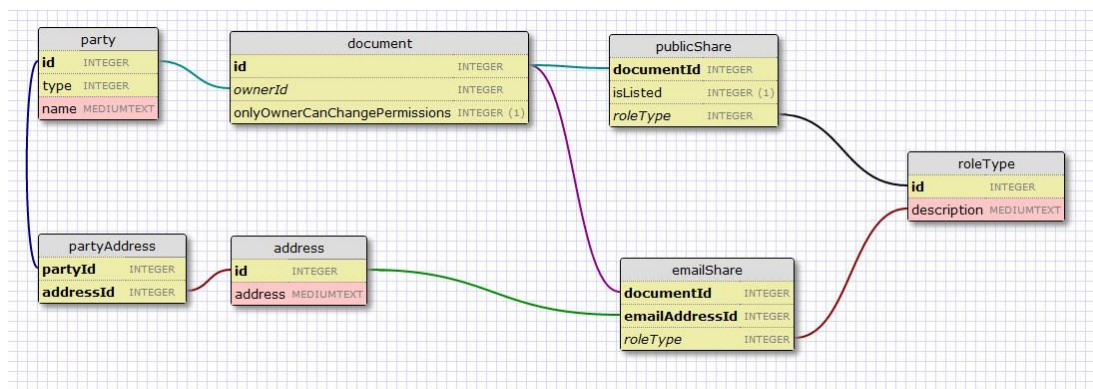


Рисунок 1.1 – Схема спільного використання файлів у хмарному сховищі «Google Drive»

Власник може регулювати публічну видимість файлу чи папки. Право власності передається. Файли або папки можуть надаватися приватним користувачам, які мають обліковий запис Google, використовуючи їхні електронні адреси @ gmail.com. Спільний доступ до файлів з користувачами, які не мають облікового запису Google, вимагає зробити їх доступними для "когось із посиланням". Це створює секретну URL-адресу для файлу, якою можна ділитися електронною поштою або приватними повідомленнями. Файли та папки також можуть бути оприлюднені в Інтернеті, а це означає, що вони можуть бути проіндексовані пошуковими системами, і таким чином їх може знайти та отримати доступ будь-хто. Власник може також встановити рівень доступу для регулювання дозволів. Три пропоновані рівні доступу - "редагувати", "коментувати" та "переглядати". Користувачі з доступом до редагування можуть запросити інших редагувати.

Результати пошуку можна звзвити за типом файлу, власністю, видимістю та відкритим додатком. Користувачі можуть шукати зображення, описуючи або називаючи те, що в них є. Наприклад, пошук "гори" повертає всі фотографії гір, а також будь-які текстові документи про гори. Текст у зображеннях та PDF може бути вилучений за допомогою оптичного розпізнавання символів. У вересні 2016 року Google додав "обробку натуральної мови" для пошуку на веб-сайті Диска Google, що

дало змогу отримати конкретні пошукові запити користувачів, наприклад "знайти таблицю мого бюджету з грудня минулого року".

Приклад завантаженого проекту в хмарному сховищі «Google Drive» має наступний вигляд (рисунок 1.2):

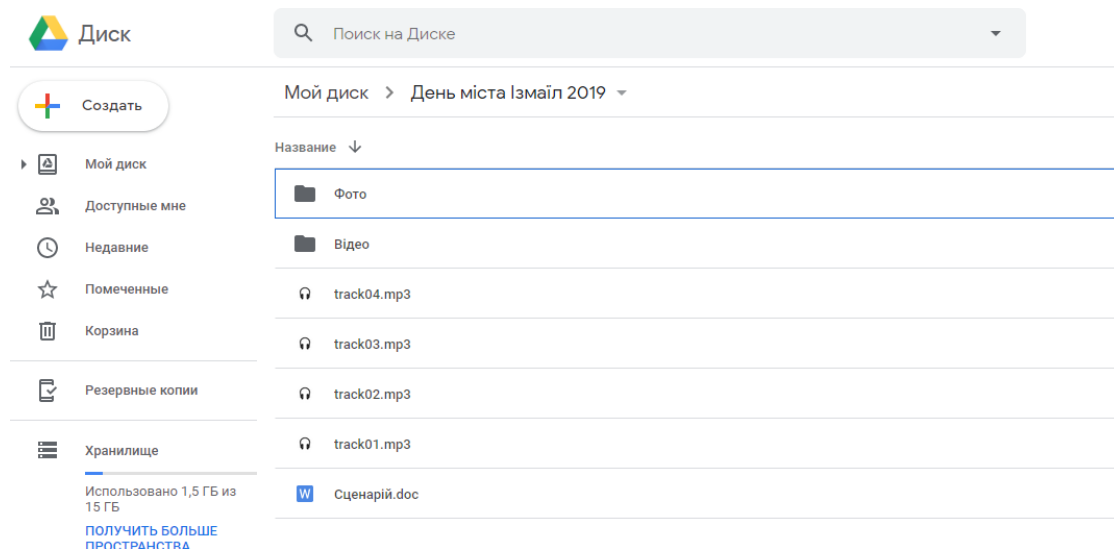


Рисунок 1.2 – загальний вигляд завантаженого організаційного заходу «День міста Ізмаїл 2019» у хмарному сховищі «Google Drive».

1.1.2 Dropbox

Dropbox - служба хостингу файлів, керована американською компанією Dropbox Inc, яка пропонує хмарне зберігання, синхронізацію файлів, персональну хмару та клієнтське програмне забезпечення.[2]

Dropbox об'єднує файли в одне центральне місце, створюючи спеціальну папку на комп'ютері користувача. Вміст цих папок синхронізується на сервери Dropbox та на інші комп'ютери та пристрої, де користувач встановив Dropbox, постійно підтримуючи однакові файли на всіх пристроях. Dropbox пропонує комп'ютерні програми для комп'ютерів Microsoft Windows, Apple macOS та Linux, а також мобільні додатки для смартфонів і планшетів iOS, Android та Windows Phone

Dropbox має комп'ютерні програми для комп'ютерів Microsoft Windows, Apple macOS та Linux, та мобільних додатків для смартфонів та планшетів iOS, Android та Windows Phone. Він також пропонує інтерфейс веб-сайту.

Програмне забезпечення Dropbox дозволяє користувачам скидати будь-який файл у визначену папку. Потім файл автоматично завантажувється в хмарну службу

Dropbox і стає доступним для будь-яких інших комп'ютерів та пристроїв користувача, на яких також встановлено програмне забезпечення Dropbox, постійно оновляючи файл у всіх системах. Коли файл у папці Dropbox користувача змінюється, Dropbox завантажує лише ті фрагменти файлу, які були змінені, коли це можливо.

Коли файл або папку видалено, користувачі можуть відновити їх протягом 30 днів. Для користувачів Dropbox Plus цей час відновлення можна продовжити до одного року, придбавши доповнення "Розширена історія версій". [3]

Dropbox також пропонує функцію синхронізації локальної мережі, де замість отримання інформації та даних із серверів Dropbox комп'ютери в локальній мережі можуть обмінюватися файлами безпосередньо між собою, потенційно значно покращуючи швидкість синхронізації.

Приклад завантаженого проекту в хмарному сховищі «Dropbox» має наступний вигляд (рисунок 1.3):

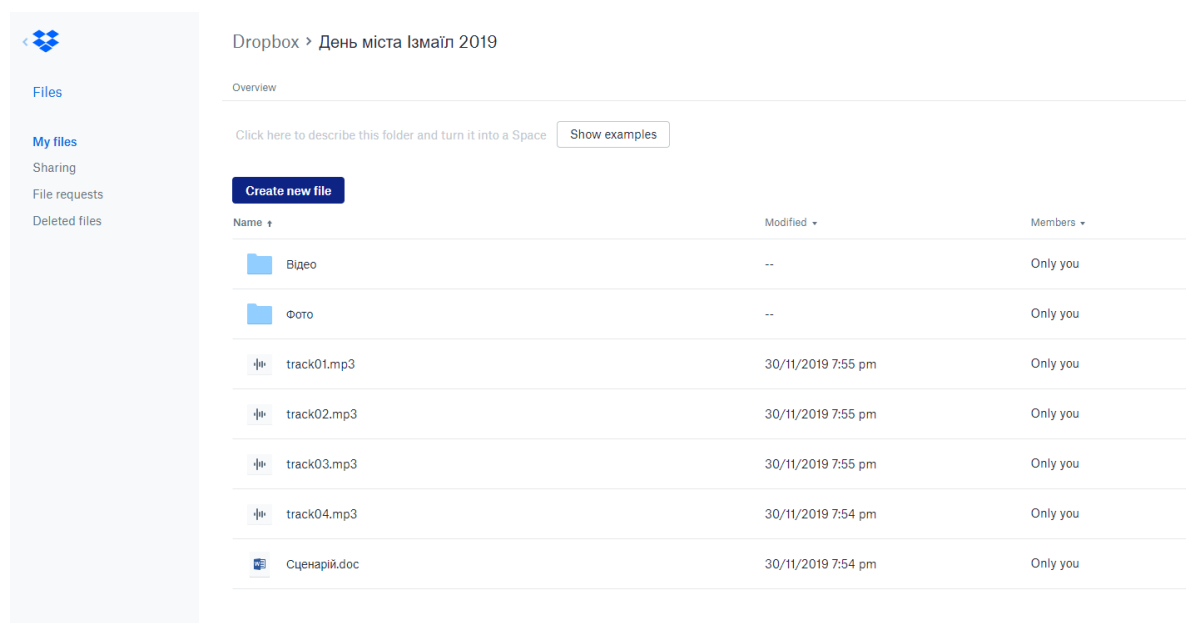


Рисунок 1.3 – загальний вигляд завантаженого організаційного заходу «День міста Ізмаїл 2019» у хмарному сховищі «Dropbox».

Було розглянуто лише 2 сервіси хмарних сховищ, але цього достатньо щоб виділити основні переваги таких платформ, адже більшість з них працює за доволі простою структурою, наведеною нижче (Рисунок 1.4).

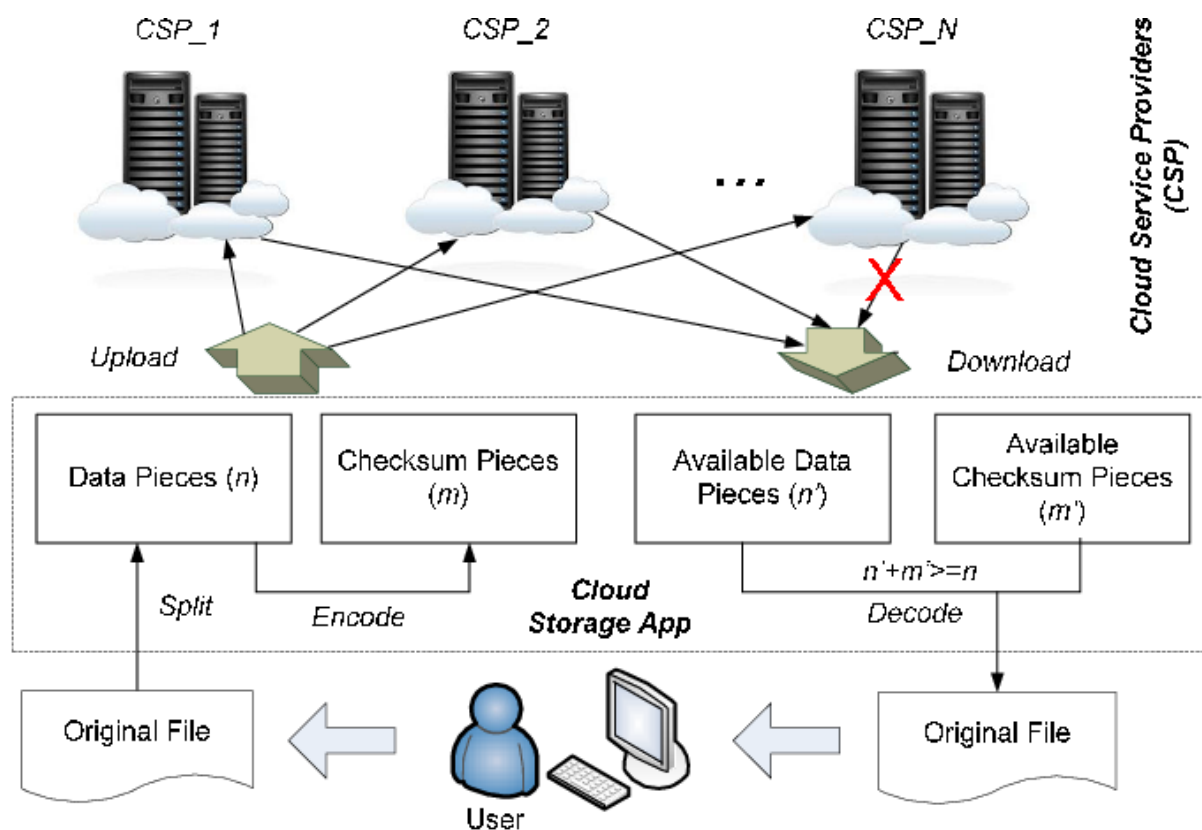


Рисунок 1.4 – структурна схема хмарних сховищ.

Основні переваги та недоліки використання хмарних сховищ в якості збереження організаційних заходів наведені нижче.

Переваги:

- необмежений об'єм збережених даних;
- легкість доступу з будь-якого пристрою;
- зручна робота декількох людей над одними файлами (навіть в один і той самий час);
- висока безпечність від вірусів;
- висока надійність збереження інформації.

Недоліки:

- необхідно мати постійний доступ до мережі Інтернет;

- незручний інтерфейс для збереження проектів.

1.2 Огляд месенджерів та соціальних мереж в якості збереження даних

1.2.1 Месенджер Telegram

Telegram - це хмарне обмін миттєвими повідомленнями та послуга передачі голосу через IP. Клієнтські програми Telegram доступні для Android, iOS, Windows Phone, Windows NT, macOS та Linux. Користувачі можуть надсилати повідомлення та обмінюватися фотографіями, відео, наклейками, аудіо та файлами будь-якого типу.

Клієнтський код Telegram - це програмне забезпечення з відкритим кодом, але вихідний код для останніх версій не завжди опублікується одразу, тоді як його код на стороні сервера є закритим та власним. Сервіс також надає API незалежним розробникам. [4]

В контексті збереження інформації в месенджері Telegram найоптимальнішим рішенням буде створення боту. Для цього необхідно буде скористатися інструкцією нижче:

1. Вводимо в пошуку «BotFather».
2. Прописуємо команду «/newbot», вводимо назву та унікальне ім'я для нашого боту.
3. Переходимо за посиланням, що згенерував бот «BotFather».
4. Вводимо команду «/start».
5. Готово.

Або можна скористатися графічною інструкцією (рисунок 1.5):

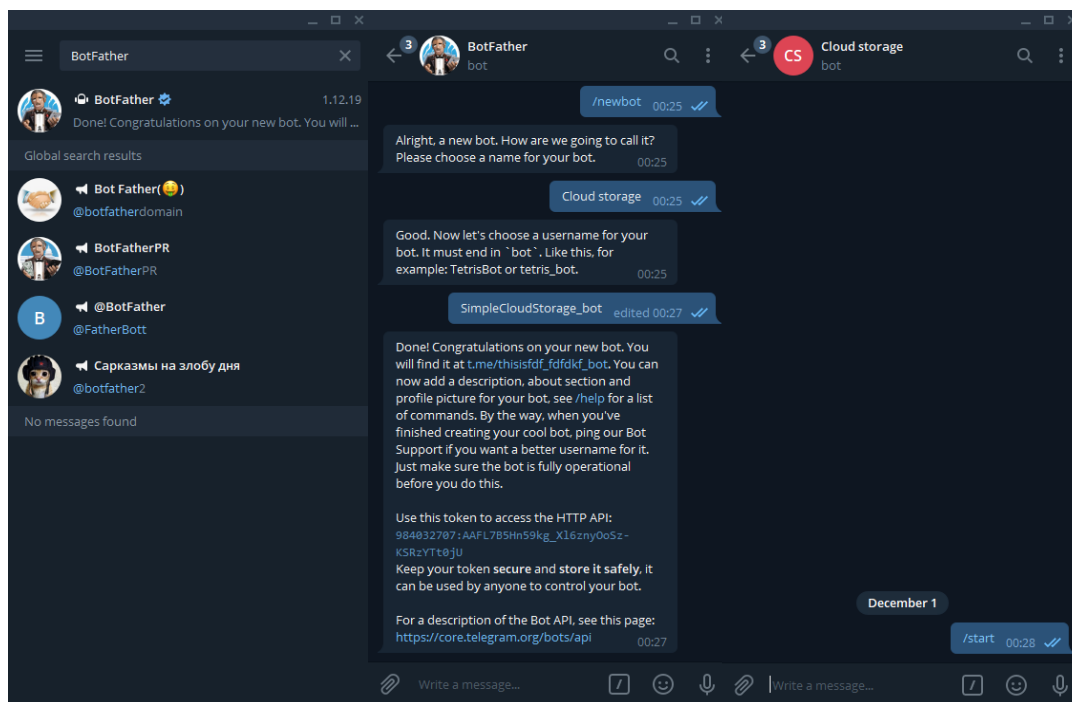


Рисунок 1.5 – графічна інструкція створення телеграм боту для збереження заходів

Кінцевий вигляд організаційного заходу в месенджері «Telegram» (рисунок 1.6):

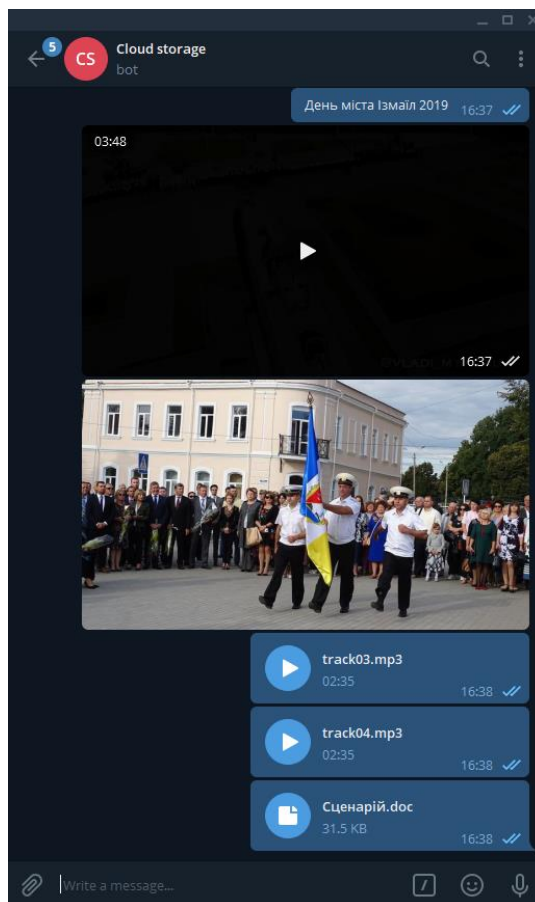


Рисунок 1.6 – приклад проекту у месенджері «Telegram»

1.2.2 Соціальна мережа VK

VK.com – найпопулярніша соціальна мережа в країнах СНГ, створена Павлом Дуровим у 2006 році. [5] VK дозволяє користувачам відправляти повідомлення один одному, створювати групи, події, ділитися та тегувати зображення, аудіо та відео.

При розгляданні нашої задачі доцільно буде скористатися функцією створення групи у даній соціальній мережі.

Приклад посту в такій групі матиме вигляд (рисунок 1.7):

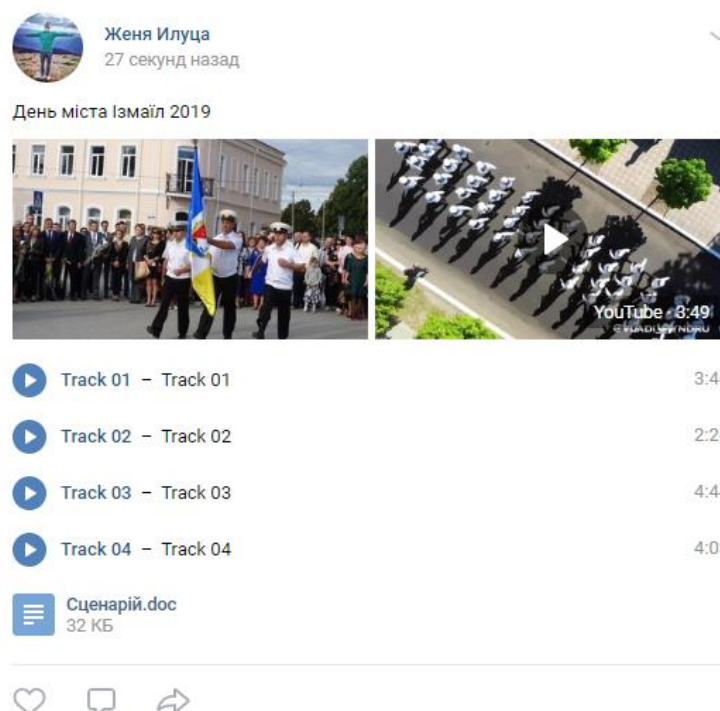


Рисунок 1.7 – приклад організаційного заходу «День міста ізмаїл 2019» в соціальній мережі

Основні переваги й недоліки вирішення нашої задачі за допомогою месенджерів або соціальних мереж наведені нижче.

Переваги:

- легкість доступу з будь-якого пристрою;
- зручність навігації;
- висока безпечність від вірусів;
- висока надійність збереження інформації;

Недоліки:

- необхідно мати постійний доступ до мережі Інтернет;
- небезпечність доступу до власних файлів;

- відсутність можливості зробити кастомну панель навігації (наприклад зміст) по проекту.

1.3 Огляд локального збереження заходів на комп'ютері

На сьогоднішній день, мабуть, найпоширеніший тип зберігання інформації. Коли йде розмова про локальне збереження маються на увазі жорсткі диски, що можуть бути внутрішніми (встановлюються всередину корпусу) і зовнішніми (приєднуються до пристрою за допомогою USB-кабелю). В останньому випадку жорсткий диск має розміри, що дозволяють носити його в кишені піджака і підключати його практично до будь-якого комп'ютера в USB-роз'єм.

Користувач має можливість зберігати на цих жорстких дисках уся необхідну йому інформацію у будь-якому форматі. Для цього не потрібно заводити акаунти та деось за щось платити. Дуже поширено серед початкових користувачів, або тих кому не потрібен миттєвий доступ до цих файлів із будь-якого пристрою в будь-який час.

До того при акуратному зверненні диски справно служать більше десяти років при активному щоденному використанні. Правда, останнім часом якість обладнання залишає бажати кращого, так як в гонитві за низькою ціною виробники економлять на обладнанні і матеріалах.

Візуальний вигляд буде схожий з виглядом проекту у хмарному сховищі, а тобто ієрархічної залежності директорій наступним чином (рисунок 1.8):



Рисунок 1.8 – ієрархічний вигляд організаційного заходу «День міста Ізмаїл 2019»

Основні переваги й недоліки вирішення нашої задачі за допомогою локального збереження файлів наведені нижче.

Переваги:

- безпечність від інтернет-загроз;
- відсутність потреби до виходу в мережу Інтернет;
- відновлення даних;
- захищеність інформації шифруванням;

Недоліки:

- ймовірність зараження вірусами від портативних носіїв доволі велика;
- схильність до несправності через неправильну;
- відсутність переглянути інформацію за допомогою різних пристроїв.

1.4 Архівування організаційних заходів на паперових носіях

Папір - найстаріший засіб зберігання інформації. Як відомо, мимовільна зміна властивостей паперу в результаті старіння пов'язане зі зміною хімічної структури і, зокрема, її основного компонента - целюлози. Розвиток технологій позитивно позначилося на якості використовуваних у виробництві матеріалів. Нові технологічні процедури дозволили значно поліпшити фізичні, хімічні і електростатичні властивості паперу. Науковий прогрес також призвів до появи більш просунутих способів нанесення інформації: чорнило на основі сажі і пір'я, грифельні олівці, авторучки, друкарська фарба, стрічки для друкарських машинок і фарби для принтера. Спосіб нанесення інформації, так само як і якість самого матеріалу, в кінцевому підсумку визначають довготривалість зберігання даних на папері. Наші предки записували букви грифелем або чорнилом на основі вуглецю, який не змінює свої властивості століттями і є хімічно стійким речовиною. Текст зазвичай наносився за допомогою фізичного пошкодження поверхні - методом продавлювання. За такою ж технологією працювали друкарські машинки і матричні принтери, в яких неорганічні барвники розпорошувалися контактним способом: спочатку папір продавлюється, а потім барвник проникав в матеріал на задану глибину. Цей старий спосіб нанесення інформації за допомогою механічного продавлювання непорівнянний з тим, що сьогодні пропонують комп'ютерні технології, але досі є дуже поширеним. Отож, людина, що зацікавлена в даній задачі, а саме збереженні або архівуванні публічних заходів повинна зробити набагато більше ніж при усіх

попередніх розглянутих способах, а саме: знайти необхідну кількість паперу, мати принтер в якому йому треба буде роздрукувати сценарій, фотографії та розташувати ці папери по каталогах, в зручному вигляді.

Основні переваги й недоліки вирішення нашої задачі за допомогою локального збереження файлів наведені нижче.

Переваги:

- доступність;

Недоліки:

- легко втратити дані;
- важкість в оформленні;
- важкість доступу даних;
- відсутність можливості додати медіа файли;
- незручність навігації.

1.5 Побудова характеристичного порівняння існуючих рішень

Було розглянуто 4 основних існуючих рішення задачі збереження публічних заходів. Порівняльний аналіз кожного з них за різними критеріями зручно спостережувати на таблиці 1.1

Таблиця 1.1 – порівняльний аналіз існуючих рішень

№ п/п	Характеристичний критерій	Кількість балів отриманих кожним із існуючих рішень			
		Cloud storage	Соціальні мережі	Локальне збереження	Паперове збереження
1.	Ціна	0 балів	1 бал	2 бали	3 балів
2.	Безпечність від інтернет-загроз	2 бали	1 бал	0 балів	3 бали
3.	Відновлення даних	4 бали	1 бал	1 бал	0 балів
4.	Зручність оформлення	2 бали	1 бал	1 бал	0 балів

Продовження таблиці 1.1

5.	Зручність користування	2 бал	1 бал	1 бал	0 балів
6.	Легкість доступу	4 бали	1 бал	1 бал	0 балів
7.	Редагування даних	2 бали	1 бали	1 бали	0 балів
8.	Конфіденційність	3 бали	1 бал	1 бал	1 бал
9.	Рейтингування	0 балів	0 балів	0 балів	0 балів

Підсумовуючи, можна з легкістю сказати, що завантаження публічних заходів у хмарне сховище є найкращим варіантом, але також, що існуючих рішень не вистачає для покриття усіх функціональних вимог користувачів. Саме тому, було вирішено створити новий програмний продукт, що буде ліпший за усі існуючі рішення у сфері управління публічними заходами.

1.6 Опис основних недоліків та побудова необхідних функціональних вимог

Після розгляду систем, що були зазначені вище, стає зрозуміло, що необхідно створити нову платформу, яка буде більш зручною для користувачів та в якій буде можливість реалізації більшої кількості задач, що стосується управління контентом публічних задач, а саме:

- зручна навігація по змісту заходу;
- можливість мати перегляд рейтингу заходу;
- сортувати заходи за типом;
- мати можливість миттєвого перегляду медіафайлів заходу;
- детальний перегляд заходу.

Висновки до розділу:

На першому етапі розробки програмного продукту системи управління контентом публічних заходів було розглянуто основні вимоги до даних систем,

описано та проаналізовано існуючі рішення даної задачі, а також описано їх недоліки та побудовано нові функціональні вимоги до системи.

2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ ТА ОПИС ФУНКЦІОНАЛЬНИХ ВИМОГ СИСТЕМИ

2.1 Вступ

У попередньому розділі було розглянуто основні існуючі рішення в задачі управління контентом публічних заходів, а також виявлено основні відмінності від нової системи, що буде розроблена. На даному етапі ми спроектуємо функціональні вимоги системи, опишемо вхідні, вихідні дані та структуру бази даних.

2.2 Опис функціональної моделі

У системі можуть існувати 3 типи акторів: неавторизований користувач, авторизований користувач, адміністратор. У кожного з них буде окремий перелік функцій. Усі функції описані у таблиці 2.1, а загальний алгоритм системи наведений у рисунку 2.1

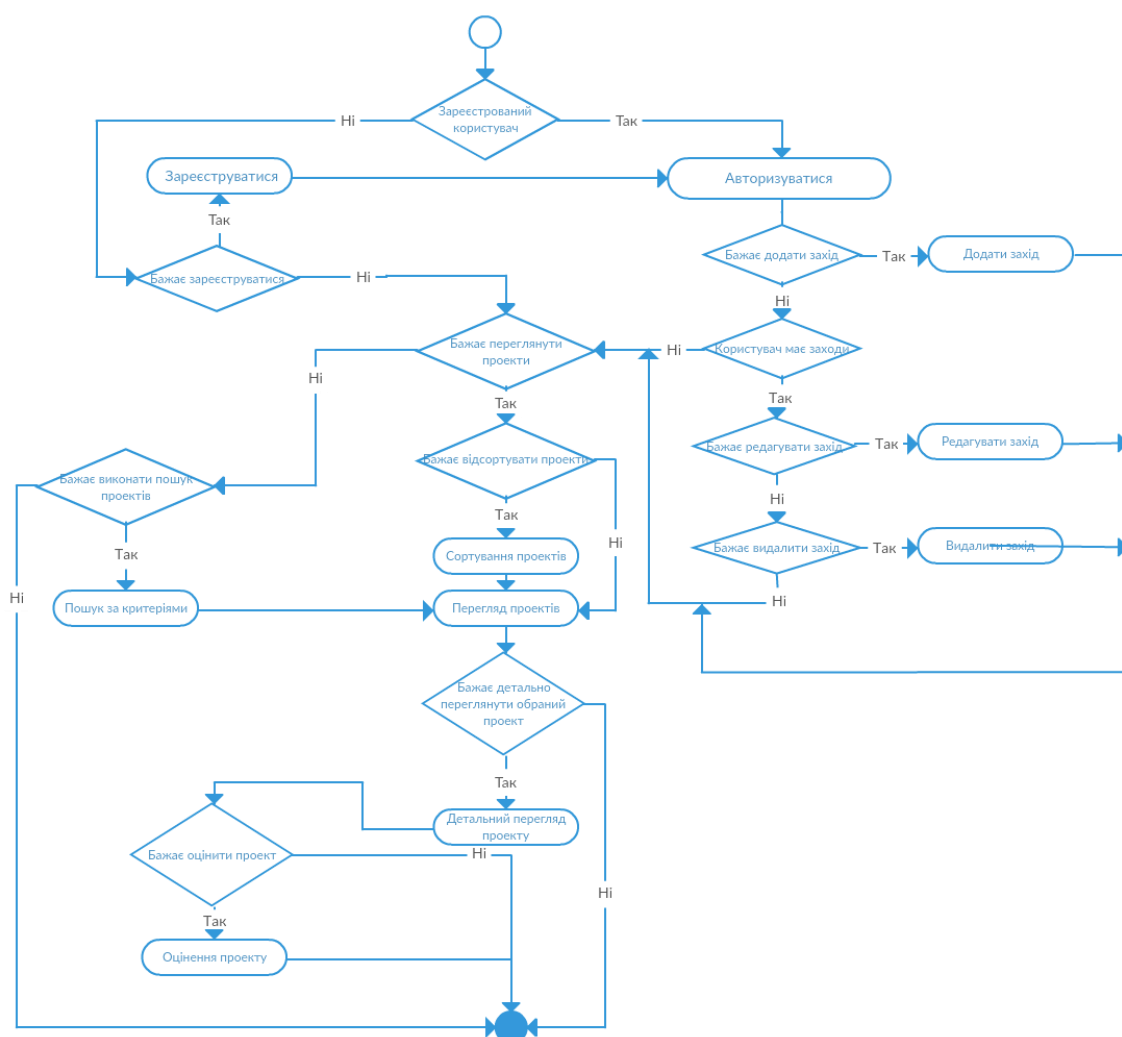


Рисунок 2.1 – Алгоритм роботи системи

Таблиця 2.1 – Опис функцій застосунку

Актор	Варіант використання	Опис дії варіанта використання
	Авторизація	Неавторизований користувач має змогу увійти в систему
	Реєстрація	Неавторизований користувач має змогу зареєструватися.
	Перегляд проектів	Неавторизований користувач має змогу бачити сторінку із усіма проектами.
	Детальний перегляд проекту	Неавторизований користувач має змогу переглянути сценарій та потрібні медіафайли обраного проекту.
	Пошук проекту	Неавторизований користувач має змогу виконати пошук по усім проектам в системі
	Сортування проекту	Неавторизований користувач має змогу зробити сортування за обраними критеріями
Авторизований користувач	Вихід із системи	Авторизований користувач має змогу вийти з системи

Продовження таблиці 2.1

Авторизований користувач	Додати проект	Авторизований користувач має змогу додати власний проект до веб-застосування
	Редагування проекту	Авторизований користувач має змогу редагувати власний проект
	Видалення проекту	Авторизований користувач має змогу видалити власний проект
	Перегляд проекту	Авторизований користувач має змогу бачити сторінку із усіма заходами.
	Оцінення проекту	Авторизований користувач має змогу поставити позначку «сподобалось» або «не сподобалось»
	Детальний перегляд проекту	Авторизований користувач має змогу переглянути сценарій та потрібні медіафайли обраного заходу.
	Пошук проектів	Авторизований користувач має змогу виконати пошук по усім проектам в системі
	Сортування проектів	Авторизований користувач має змогу зробити сортування за обраними критеріями

Продовження таблиці 2.1

Адміністратор	Додати проект	Адміністратор має змогу додати власний проект до веб-застосування
	Редагування проекту	Адміністратор має змогу редагувати будь-який проект
	Видалення проекту	Адміністратор має змогу видалити будь-який проект
	Перегляд проекту	Адміністратор має змогу бачити сторінку із усіма заходами.
	Оцінення проекту	Адміністратор має змогу поставити позначку «сподобалось» або «не сподобалось»
	Детальний перегляд проекту	Адміністратор має змогу переглянути сценарій та потрібні медіафайли обраного заходу.
	Пошук проектів	Адміністратор має змогу виконати пошук по усім проектам в системі
	Сортування проектів	Адміністратор має змогу зробити сортування за обраними критеріями
	Вихід із системи	Авторизований користувач має змогу вийти з системи

2.3 Вхідні дані

Для правильного функціонування системи вхідні дані, мають бути наступними:

- проект – при доданні проекту користувач повинен дати на вхід в систему наступну інформацію: назва, короткий опис проекту, сценарій проекту у розширенні doc/docx та за бажанням медіа матеріали;
- дані про авторизованих користувачів – при реєстрації користувачу повинен ввести такі дані як: ім'я, фамілія, логін, пошта та пароль.

2.4 Вихідні дані

На вихід від програмного продукту, можуть отримуватися наступні дані:

- проект, збережений в ручному вигляді, із можливістю перегляду рейтингу, а також миттєвий відтворюванням медіа контенту.

2.5 Опис структури бази даних

Публічна схема бази даних складатиметься усього із трьох таблиць: User, Project, Content.

Опис сутностей наведено у таблиці 2.1.

Таблиця 2.1 – Опис сутностей бази даних

Назва	Поле	Призначення
	[Id](bigserial)	Унікальний ідентифікатор
	[First_Name] (text)	Ім'я
	[Second_Name] (text)	Прізвище
	[Email] (text)	Електронна адреса
	[Password] (text)	Пароль
	[Type] (bigint)	Тип користувача
	[Id] (bigserial)	Унікальний ідентифікатор
	[Name] (text)	Назва проекту
	[DocData] (jsonb)	Сценарій проекту
	[DocLink] (text)	Посилання на проект
	[Likes] (int)	Кількість «лайків»
	[Dislikes] (int)	Кількість «дизлайків»
	[Views] (int)	Кількість переглядів
	[User_ID] (int)	Посилання на користувача в системі

Content – дані про завантажені медіафайли	[Id] (int)	Унікальний ідентифікатор
	[Link] (varchar)	Посилання на медіафайл
	[ContentType] (varchar)	Тип медіафайлу
	[Project_ID] (int)	Посилання на проект в системі

Задля того, щоб наша база даних була нормалізованою необхідно додати необхідні обмеження:

- User.id primary key constraint;
- Project.id primary key constraint;
- Content. id primary key constraint;
- Project.userid foreign key constraint references(User.id);
- Content.projectid foreign key constraint references(Project.id);
- User.type check constraint any(array['admin', 'user']);
- Project.likes check constraint (likes >= 0);
- Project.dislikes check constraint (dislikes >= 0);
- Project.views check constraint (views >= 0);
- Project.type check constraint any(array['video', 'audio']).

А також додамо індекси, що покриватимуть усі наші запити:

- create index concurrently idx_user_email on user(email);
- create index concurrently idx_user_fullname on user(firstname, secondname);
- create index concurrently idx_project_name on Project(name);
- create index concurrently idx_project_userid on Project(userid);
- create index concurrently idx_project_createtime on Project(createtime);
- create index concurrently idx_content_projectid on Content(projectid).

Отже, після вище описаних обмежень на поля об'єктів система буде володіти трьома принциповими властивостями для цілісності бази даних (Database Integrity, Data Consistency, Referential Integrity). А після додання необхідних індексів ми оптимізували усі запити до БД, згідно функціональних вимог системи.

Загальна схема бази даних програмного забезпечення знаходиться в пояснювальній записці загального тому дипломного проекту. Схему бази даних програмного забезпечення, яка відноситься до даного модуля наведена нижче (рисунок 2.2).

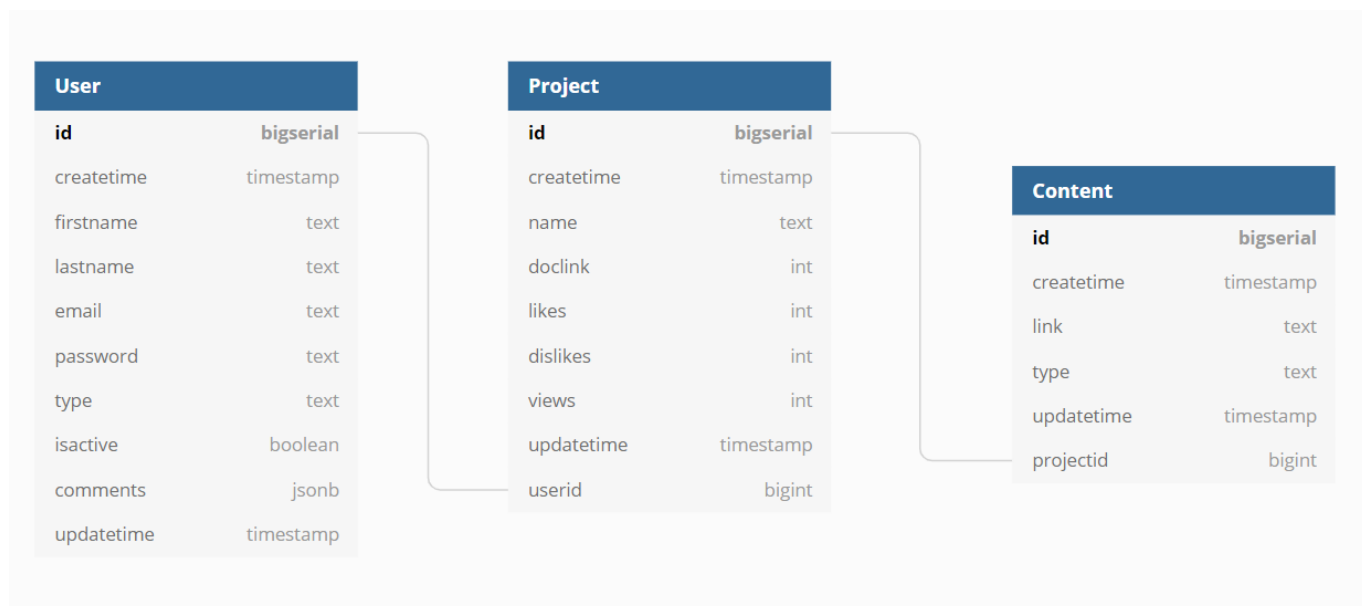


Рисунок 2.2 – Схема бази даних системи

Висновки до розділу:

У даному розділі було чітко сформовано усі функціональні вимоги системи та побудовану схему структурної діяльності, виявлено усі вхідні та вихідні дані системи, спроектована, описана та покрита усіма необхідними обмеженнями та індексами схема бази даних.

3 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ТА ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Вступ

На попередньому етапі розробки програмного забезпечення системи управління контентом публічних заходів ми сформувавши необхідні функціональні вимоги системи, а також розробили схему бази даних. Це дає нам змогу на даній стадії розробки провести аналіз існуючих засобів розробки та відповідно почати розробку веб-додатку.

3.2 Засоби розробки

3.2.1 Мова програмування

При розробці програмного продукту обов'язковою складовою є вибір мови програмування. В нашому випадку було розглянуто дві мови програмування – Ruby on Rails та Python.

Python [6] — інтерпретована об'єктно-орієнтована мова програмування високого рівня з строгою динамічною типізацією. Платформа Python зараз розвинена настільки, що дозволяють вирішувати практично всі прикладні завдання стандартним чином, які не потребують низькорівневої взаємодії з апаратурою. Python відомий гарною надійністю роботи серверних додатків, зручний механізм багатопоточності полегшує створення додатків, які обслуговують одночасно безліч користувачів.

Структури даних високого рівня разом із динамічною семантикою та динамічним зв'язуванням роблять її привабливою для швидкої розробки програм, а також як засіб поєднання існуючих компонентів. Python підтримує модулі та пакети модулів, що сприяє модульності та повторному використанню коду. Інтерпретатор Python та стандартні бібліотеки доступні як у скомпільованій так і у вихідній формі на всіх основних платформах. В мові програмування Python підтримується кілька парадигм програмування, зокрема: об'єктно-орієнтована, процедурна, функціональна та аспектно-орієнтована.

Основними перевагами мови Python є:

- чистий синтаксис (для виділення блоків слід використовувати відступи);
- переносність програм (що властиве більшості інтерпретованих мов);

- стандартний дистрибутив має велику кількість корисних модулів (включно з модулем для розробки графічного інтерфейсу);
- можливість використання Python в діалоговому режимі (дуже корисне для експериментування та розв'язання простих задач);
- стандартний дистрибутив має просте, але разом із тим досить потужне середовище розробки, яке зветься IDLE і яке написане на мові Python;
- зручний для розв'язання математичних проблем (має засоби роботи з комплексними числами, може оперувати з цілими числами довільної величини, у діалоговому режимі може використовуватися як потужний калькулятор);
- відкритий код (можливість редагувати його іншими користувачами).

Python має ефективні структури даних високого рівня та простий, але ефективний підхід до об'єктно-орієнтованого програмування. Елегантний синтаксис Python, динамічна обробка типів, а також те, що це інтерпретована мова, роблять її ідеальною для написання скриптів та швидкої розробки прикладних програм у багатьох галузях на більшості платформ. [7]

Інтерпретатор мови Python і багата Стандартна бібліотека (як вихідні тексти, так і бінарні дистрибутиви для всіх основних операційних систем) можуть бути отримані з сайту Python www.python.org, і можуть вільно розповсюджуватися. Цей самий сайт має дистрибутиви та посилання на численні модулі, програми, утиліти та додаткову документацію.

Інтерпретатор мови Python може бути розширений функціями та типами даних, розробленими на C чи C++ (або на іншій мові, яку можна викликати із C). Python також зручна як мова розширення для прикладних програм, що потребують подальшого налагодження.

Ruby - об'єктно-орієнтована мова, розроблена Юкіхіро Мацумото (також відомий як Мац у спільноті Рубі) в середині 1990-х років в Японії. В першу чергу, слід зазначити, що все в Ruby є об'єктом. Мета розробки Ruby полягала в тому, щоб він діяв як розумний буфер між людськими програмістами та базовими обчислювальними механізмами. У Ruby синтаксис схожий з багатьма мовами

програмування, такими як C і Java, тому програмістам Java та C легко вивчити. Він підтримує в основному всі платформи, такі як Windows, Mac, Linux. [8]

Ruby заснований на багатьох інших мовах, таких як Perl, Lisp, Smalltalk, Eiffel і Ada. Це інтерпретована мова, тому більшість її реалізацій виконують інструкції прямо та вільно, без попереднього компілювання програми в інструкції на машинній мові. Програмісти Ruby також мають доступ до потужних RubyGems (RubyGems забезпечує доступ бібліотек Ruby).

Він має підтримку самоаналізу, рефлексії та метапрограмування, а також підтримку потоків на основі інтерпретації. Ruby має динамічну типізацію та підтримує параметричний поліморфізм.

Переваги Ruby:

- код, написаний на Ruby, невеликий, елегантний та потужний, оскільки має меншу кількість рядків коду;
- Ruby дозволяє просте і швидке створення веб-додатків, що призводить до менш важкої роботи;
- оскільки Ruby є безкоштовним, тобто Ruby вільний копіювати, використовувати, змінювати, він дозволяє програмістам вносити необхідні зміни як і коли потрібно;
- Ruby - це динамічна мова програмування, завдяки якій немає жорстких правил щодо вбудованих функцій, і вона дуже близька до розмовної мови. [9]

Недоліки Ruby:

- Ruby є досить новим і має свою унікальну мову кодування, що ускладнює програмістам кодування в ньому відразу, але після деякої практики його просто використовувати. Багато програмістів вважають за краще дотримуватися того, що вони вже знають і можуть розвивати;
- код, написаний у Ruby, важче налагоджувати, оскільки більшість часу він генерується під час виконання, тому його важко читати під час налагодження;
- Ruby не має великої кількості інформаційних ресурсів порівняно з іншими мовами програмування;
- Ruby - інтерпретована сценарна мова, мови сценаріїв зазвичай повільніше, ніж компільовані мови, тому Ruby повільніше, ніж у багатьох інших мов.

Взагалі, Ruby використовується для створення веб-додатків різного роду. Це одна з найсучасніших технологій в даний час для створення веб-додатків. Ruby пропонує чудовий фреймворк під назвою Ruby on Rails (RoR). Це веб-структура, яка використовується програмістами для прискорення процесу розробки та економії часу.

Порівняємо обидві мови програмування – Ruby та Python. В багатьох випадках вони є дуже схожими. [10]

Таку схожість можливо спостерігати в наступних характеристиках:

- вони доступні відповідно до ліцензій, затверджених OSI та FSF, тому є абсолютно безкоштовними, щоб користуватися ними або розповсюджувати програмне забезпечення, створене даними мовами програмування;
- вони також є кросплатформенними, що зручно для розподілених команд, де люди можуть використовувати Windows або Linux на своїх комп'ютерах;
- Ruby і Python - це сценарії високого рівня; їхні програми не потрібно збирати;
- обидві мови мають нестрогу типізацію, це означає, що ви можете використовувати змінну, не оголошуючи її тип;
- обидві мови підтримують об'єктно-орієнтоване програмування;
- обидві підтримують функції Lambda у веб-службах Amazon (AWS).

Але відмінностей в них набагато більше. Розберемо кожен з них.

Сфери використання.

Ruby поширений в організаціях, які створюють веб-додатки. Програмування за допомогою Ruby on Rails дозволяє командам швидко розвиватись і зосереджуватися на бізнес-процесах замість написання функцій з нуля. Ruby забезпечує елегантну структуру MVC, тому ви можете розділяти свої дані, інтерфейс користувача та бізнес-функції. [11]

Django - найпопулярніша структура MVC в Python, але Python також відомий поза сферою веб-додатків. Прикладом може бути бібліотека Pandas, яка корисна для обробки даних, а також інших математичних бібліотек, таких як numpy. TensorFlow популярний для завдань машинного навчання, а Matplotlib є потужним для візуалізації даних. Також варто згадати SciPy, який має ресурси для вирішення математичних функцій. Отож сфер використання у Python значно більше.

Гнучкість.

У цьому аспекті Ruby успадкувала філософію Perl: "Існує більше ніж один спосіб зробити одну й ту саму задачу". Отже, ви завжди знайдете багато різних методів для досягнення вирішення певного завдання в Ruby. Залежно від того, хто пише код, це може призвести до зайвої складності та заплутаності.

Python ж навпаки дотримується підходу, коли «простота має більше значення, ніж складність» (кн. Python`s Zen). Отже, його філософія полягає в тому, що "повинен бути один - і бажано лише один - найпростіший спосіб зробити це". Отже, хоча код Python, ймовірно, не буде найбільш гнучким, він має хороші шанси бути більш читабельним у випадку колективної розробки.

Популярність.

Дивлячись на Octoverse 2018 від GitHub, ми можемо побачити, як використання Ruby повільно занепадає в топ-10 мов, з п'ятого місця в 2014 році до десятого місця в 2018 році. Однак Python стабільно зберігає свою популярність, переходячи з четвертої найбільш вживаної мови в 2014 році вгору до третього місця у 2015 році, де він залишився до 2018 року.

Давайте перевіримо пропозиції вакансій як ще один критерій: пошук віддалених завдань у Stack Overflow дає результати вдвічі більше, коли ви шукаєте Python замість Ruby. Якщо ви віддаєте перевагу LinkedIn, Python лідирує на 20% над Ruby.

Код багаторазового використання.

Загальнодоступний і готовий до використання код є важливим фактором, коли потрібно визначитися з мовою програмування. Python називає їх "модулями", і вони доступні через PyPI, де ви можете знайти більше 150 000 модулів. Аналог модулів у Ruby є Gems, їх приблизно стільки ж. Але диференціюючим фактором є фільтрація; PyPI дозволяє фільтрувати за такими категоріями, як "статус розробки", що простіше, ніж порівнювати багато бібліотек та оцінювати їх код вручну.

Підсумовуючи, слід навести порівняльний аналіз Python та Ruby за обраними критеріями (таблиця 3.2) та зробити вибір мови програмування, що буде використовуватися при розробці програмного продукту.

Таблиця 3.2 – характеристичне порівняння MySQL та PostgreSQL

№ п/п	Характеристичний критерій	Python	Ruby	Обґрунтування
1.	Популярність	+	-	Має набагато більшу популярність за індексом TIOBE – 10.3% Python проти 1.3% Ruby
2.	Легкість використання	+	-	Python набагато більш зручний за допомогою стандарту форматування PEP-8, а також значно зручніший у налагодженні помилок через відсутність багатьох способів розв'язання задачі
3.	Сфери використання	+	-	Python на відміну від Ruby може використовуватись у таких сферах діяльності як: Machine Learning та Data Analysis
4.	Швидкість	+	-	Python швидше Ruby в більшості випадків в силу того, що на відміну від другого, перший конвертує абстрактне синтаксичне дерево (Abstract Syntax Tree, AST) у байт-код тільки один раз, а не в режимі «реального часу» за необхідністю.

Продовження таблиці 3.2

5.	Кросплатформеність	-	+	Ruby має повну сумісність із мобільною розробкою, без необхідності налаштування сторонніх фреймворків
6.	Підтримка	+	-	Python має значно більшу та значно сильнішу підтримку у порівнянні із Ruby через свою популярність

Із вищезазначеного робимо висновок про необхідність надання переваги мови програмування Python.

3.2.2 Середовище розробки (IDE):

PyCharm - на сьогодні одне із найоптимальніших середовищ розробки для мови програмування Python. Має багато функцій, що полегшують рутинну роботу програміста. PyCharm, в даному середовищі розробки, одне з кращих додатків для завершення коду серед існуючих аналогів, що відіграє важливу роль у швидкому та якісному написанні коду.

3.2.3 Система керування базами даних

При виборі СКБД системи для розроблення програмного продукту системи збереження публічних заходів, було розглянуто дві найліпші безкоштовні СКБД: PostgreSQL та MySQL. Отож, доцільно буде їх розглянути, проаналізувати та порівняти одну з іншою.

PostgreSQL — компактний багатопотоковий сервер баз даних. Характеризується високою швидкістю, стійкістю і простотою використання.

PostgreSQL вважається гарним рішенням для малих і середніх застосувань. Сирцеві коди сервера компілюються на багатьох платформах. Найповніше можливості сервера виявляються в UNIX-системах, де є підтримка багатопоточності, що підвищує продуктивність системи в цілому. [14]

Можливості сервера PostgreSQL:

- підтримується необмежена кількість користувачів, що одночасно працюють із БД;
- висока швидкість виконання команд;
- наявність простої і ефективної системи безпеки.
- простота у встановленні та використанні.[15]

MySQL – це безкоштовна система керування реляційними базами даних (RDBMS), яка використовується для розробки веб-програмних програм. Була розроблена Майклом Віденусом та Девідом Ексмарком у 1994 році на мові C та C ++. Відзначається своєю швидкістю, надійністю, гнучкістю та простотою у користуванні особливо поширена у невеликих компаніях. В даний час MySQL розробляється, розповсюджується та підтримується корпорацією Oracle. [16]

MySQL отримав позитивні відгуки, і рецензенти помітили, що це "надзвичайна швидка СКБД в програмах середньої важкості. Він також був перевірений як "швидкий, стабільний і справжній багатокористувацький багатопотоковий сервер бази даних SQL".

Можливості сервера PostgreSQL:

- сервер MySQL є багат шаровим з незалежними модулями;
- повністю багатопоточний. Він може працювати з декількома процесорами, якщо вони доступні;
- забезпечує транзакційні та не транзакційні системи зберігання;
- має високошвидкісну систему розподілу пам'яті на основі потоку;
- підтримує таблицю купи пам'яті;
- працює на багатьох різних платформах.

Особливості PostgreSQL

- активна громада, яка прискорює свій розвиток;
- найпоширеніша альтернатива Oracle, DB2 та SQL Server;
- працює на всіх основних платформах ОС, які у вас можуть бути;
- MVCC підтримує велику кількість одночасних користувачів;
- широка індексація для високоефективної звітності;
- підтримка сучасних додатків (XML та JSON);

- підтримка ANSI SQL для переносних навичок / коду;
- зовнішні ключі підтримують ефективне зберігання даних;
- приєднання таблиці та представлення даних для гнучкого пошуку даних;
- тригери / збережені процедури для складних програм та транзакцій;
- реплікація для резервного копіювання даних та масштабованості читання.

Особливості MySQL

- MySQL - це СКБД, керована спільнотою;
- сумісний з різними платформами з використанням усіх основних мов та програмного забезпечення;
- він пропонує підтримку багатOVERСІЙНОГО контролю одночасності;
- сумісний зі стандартом ANSI SQL;
- дозволяє на основі журналу та на основі тригера реплікації SSL;
- об'єктно-орієнтовані та ANSI-SQL2008 сумісні;
- багатошаровий дизайн з незалежними модулями;
- повністю багатопотоковий, використовуючи Kernel;
- сервер доступний у вбудованій моделі DB або клієнтського сервера;
- пропонує вбудовані інструменти для аналізу запитів та аналізу простору;
- MySQL працює на багатьох різновидах UNIX, а також на інших системах, що не є UNIX, такими як Windows та OS / 2.

Більш детально порівняти ці дві СКБД більш доцільно в таблиці 3.1

Таблиця 3.1 – характеристичні відмінності MySQL та PostgreSQL

Параметр	MySQL	PostgreSQL
Open Source	Проект MySQL надав свій вихідний код доступним згідно з умовами Загальної публічної ліцензії GNU.	PostgreSQL випускається під ліцензією PostgreSQL, що є безкоштовною ліцензією Open Source. Це схоже на ліцензії BSD & MIT.

ACID	MySQL сумісний з ACID лише тоді, коли він використовується з двигунами InnoDB та NDB Cluster Storage.	PostgreSQL є повністю сумісним з ACID.
Сумісність із SQL-стандартами	MySQL частково сумісний з SQL-стандартами. Наприклад, він не підтримує constraint check	PostgreSQL повною мірою сумісний з SQL-92, SQL-99 та SQL-2004
Підтримка ком'юніті	У ньому є велика кількість співавторів, які зосереджуються в основному на підтримці існуючих функцій з новими функціями, що з'являються дуже рідко.	Активна спільнота постійно вдосконалює існуючі функції, тоді як її інноваційна спільнота прагне забезпечити, щоб вона залишалася найдосконалішою базою даних. Регулярно випускаються нові передові функції та покращення безпеки.

Продовження таблиці 3.1

Продуктивність	Використовується в основному для веб-проектів, яким не потрібна різні типи ізоляваності транзакцій	Він широко використовується у великих системах, де важливі швидкості читання та запису
Підтримка JSON	підтримує тип даних JSON, але не підтримує жодну іншу функцію NoSQL.	Підтримка JSON та інших функцій NoSQL, таких як рідна підтримка XML. Це також дозволяє індексувати дані JSON для більш швидкого доступу.
Підтримка матеріалізованих представлень	Підтримка тільки тимчасових таблиць.	Підтримка матеріалізованих подань та тимчасових таблиць, але є необхідність в ручному оновленні представлення.
Екосистема	MySQL має динамічну екосистему з такими варіантами, як MariaDB, Percona, Galera тощо.	Postgres має обмежені можливості високого класу. Однак він змінюється новими можливостями, представленими в останній версії.
B-tree індекси	Два або більше B-tree індексів можна використовувати, коли це доречно.	B-tree індекси об'єднані між собою, тобто є динамічно перетвореними предикатами.

Продовження таблиці 3.1

Статистика об'єктів	Досить хороша статистика об'єктів	Дуже хороша статистика об'єктів
Зірок в GitHub	3.34 тис.	5.6 тис.
Питань на Stack Overflow	532К	89.3К

Недоліки PostgreSQL:

- сучасні рішення вимагають високої кваліфікації співробітників;
- важкість оновлення версії для великих релізів;
- під час оновлення потрібне подвійне зберігання;
- плани виконання запитів не кешовані;
- операції масового завантаження можуть стати дуже вимогливими до процесору.

Недоліки MySQL:

- операції, пов'язані із системним каталогом, не відповідають стандартам ACID;
- немає модуля аутентифікації, що запобігає централізованому керуванню обліковим записом;
- немає підтримки ролей, тому важко підтримувати привілеї для багатьох користувачів;
- збережені процедури не підлягають кешуванню;
- таблиці, які використовуються для процедури або тригера, завжди попередньо блокуються.

Ключові відмінності між MySQL та PostgreSQL:

- PostgreSQL підтримує усі сучасні текстові формати обміну даними, такі як JSON, XML тощо, тоді як MySQL підтримує лише JSON.
- PostgreSQL добре працює при виконанні складних запитів, тоді як MySQL добре працює в системах OLAP та OLTP.
- PostgreSQL є повністю сумісний з ACID, тоді як MySQL сумісний лише з ACID, коли використовується з InnoDB та NDB.

- PostgreSQL підтримує Materialized Views з необхідністю ручного оновлення даних, тоді як MySQL не підтримує Materialized Views взагалі. [17]

Підсумовуючи, слід навести порівняльний аналіз MySQL та PostgreSQL за обраними критеріями (таблиця 3.2) та зробити вибір СКБД, що буде використовуватися при розробці програмного продукту.

Таблиця 3.2 – характеристичне порівняння MySQL та PostgreSQL

№ п/п	Характеристичний критерій	MySQL	PostgreSQL	Обґрунтування
1.	SQL сумісність	-	+	Має набагато більшу сумісність із SQL-стандартами (SQL-94, SQL-2002, SQL-2004)
2.	Безпека	+	-	У MySQL вбудовано набагато більше функцій безпеки
3.	Реплікації	-	+	PostgreSQL може виконувати реплікацію master-slave, master-master, а також інші типи реплікацій за допомогою сторонніх розширень
4.	Швидкість	-	+	Широко застосовується у великих системах, де швидкості читання і запису є вирішальними і потребують виконання складних запитів. MySQL навпаки показує себе як ненадійною СКБД, як тільки це буде заповідіано великими навантаженнями.

Продовження таблиці 3.2

5.	Підтримка	-	+	PostgreSQL має дуже сильну та активну спільноту, яка працює над задачами різних сфер використання. MySQL велика спільнота, яка зосереджена на підтримці існуючих функцій, в той час як нові функції випускаються рідко.
----	-----------	---	---	---

Із вищезазначеного робимо висновок про необхідність надання переваги СКБД PostgreSQL.

3.2.4 Представлення

Будучи веб-орієнтованим, Django потребує зручного способу динамічного генерування HTML. Шаблон містить статичні частини потрібного виводу HTML, а також якийсь спеціальний синтаксис, що описує, як буде вставлено динамічний контент. Проект Django можна налаштувати за допомогою одного або декількох двигунів шаблонів (або навіть нуля, якщо ви не використовуєте шаблони). Django постачає вбудовані програмні засоби для власної системи шаблонів, творчо названої мовою шаблонів Django (DTL), і для популярної альтернативи Jinja2. Розширення для інших мов шаблонів можуть бути доступні від третіх сторін. Django визначає стандартний API для завантаження та візуалізації шаблонів незалежно від резервного інтерфейсу. Завантаження складається з пошуку шаблону для даного ідентифікатора та попередньої його обробки, зазвичай його компіляції для представлення в пам'яті. Візуалізація означає інтерполяцію шаблону з контекстними даними та повернення отриманого рядка.

Django Template - це власна система шаблонів Django. До Django 1.8 це був єдиний доступний вбудований варіант. Це гарна бібліотека шаблонів, навіть при роботі із високонавантаженими системами. Якщо у вас немає нагальної причини вибрати інший бекенд, слід скористатися DTL, особливо якщо ви пишете додаток, що

підключається, і ви маєте намір поширювати шаблони. Додатки програми Django, що містять шаблони, як `django.contrib.admin` також використовують DTL.

3.3 Вимоги до технічного забезпечення

Для коректної роботи даного продукту на стороні користувача мають бути такі технічні характеристики:

1) комп'ютер:

- процесор Intel Celeron із тактовою частотою 1 ГГц або відповідний йому іншого модельного ряду;
- оперативної пам'ять - не менше 512 МБ;

2) програмне забезпечення:

- операційна система Windows 98+;
- браузер з підтримкою JavaScript.

3) комп'ютерна периферія:

- монітор,
- мишка,
- клавіатура.

Для коректної роботи даного продукту на серверній стороні рекомендовані такі технічні характеристики:

1) комп'ютер:

- процесор Intel Xeon E5-2620 або йому відповідний;
- об'єм оперативної пам'яті - не менше 32 ГБ;
- накопичувач розміром 4 ТВ

2) програмне забезпечення:

- операційна система Linux із версією ядра 3.10+;
- Python 3.6;
- сервер застосувань – Apache HTTP Server 2.4.33;
- сервер бази даних PostgreSQL 9.5;

3.4 Архітектура програмного забезпечення

В ході створення програмного продукту було обрано схему MVC (Model-View-Controller). MVC - це модель створення додатків, що складається з трьох взаємопов'язаних частин. Вони включають модель (дані), вид (інтерфейс користувача) та контролер (процеси, що обробляють введення).

Модель або модель "MVC" зазвичай використовується для розробки сучасних інтерфейсів користувача. Він пропонує основні елементи для розробки програм для настільних або мобільних пристроїв, а також веб-додатків. Він добре працює з об'єктно-орієнтованим програмуванням, оскільки різні моделі, види та контролери можуть розглядатися як об'єкти та повторно використовуватися в програмі. Структурна схема MVC наведена на рисунку 3.1.

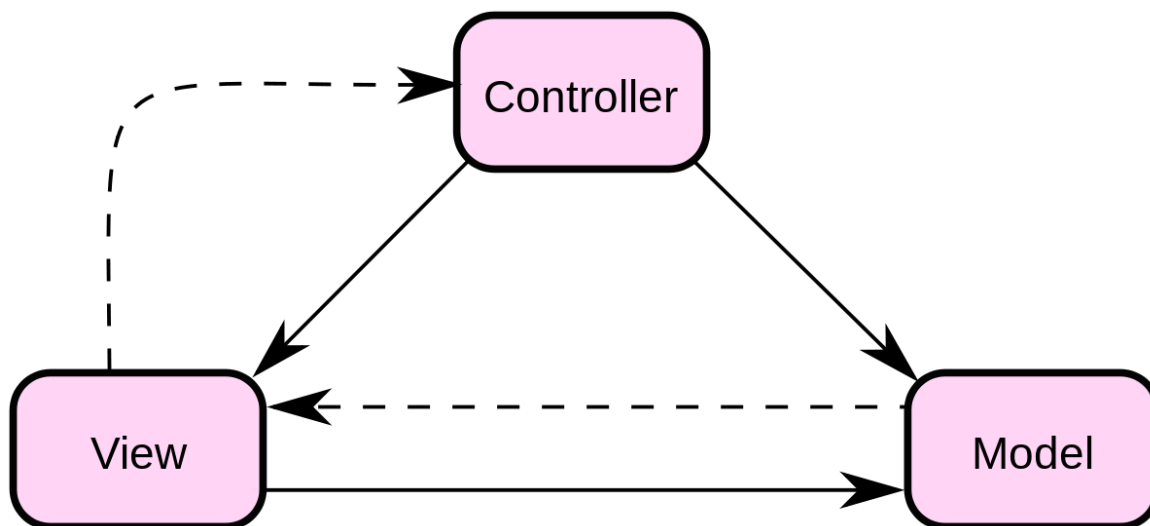


Рисунок 3.1 – Діаграма класів системи збереження організаційних заходів

3.4.1 Діаграма класів

На даному етапі розробки доцільно будет розробити «каркас» системи, та описати поля та методи кожного з класів системи (рисунок 3.1).

User	Project	Content
[id]	[id]	[id]
[firstname]	[name]	[link]
[lastname]	[doclink]	[type]
[email]	[likes]	[projectid]
[password]	[dislikes]	def __init__(self, id, link, type, projectid)
[type]	[views]	@property def link(self)
[comments]	[userid]	@link.setter def link(self)
def __init__(self, id, firstname, lastname, email, password, type)	def __init__(self, id, name, doclink, likes, dislikes, views, userid)	@property def type(self)
@property def firstname(self)	@property def name(self)	@type.setter def type(self)
@firstname.setter def firstname(self)	@name.setter def name(self)	@property def projectid(self)
@property def lastname(self)	@property def doclink(self)	@projectid.setter def projectid(self)
@lastname.setter def lastname(self)	@doclink.setter def doclink(self)	def edit(self)
@property def email(self)	@property def likes(self)	def remove(self)
@email.setter def email(self)	@likes.setter def likes(self)	
@property def password(self)	@property def dislikes(self)	
@password.setter def type(self)	@dislikes.setter def dislikes(self)	
@property def comments(self)	@property def views(self)	
@comments.setter def comments(self)	@views.setter def views(self)	
def deactivate(self)	@property def userid(self)	
	@userid.setter def userid(self)	
	def edit (self)	
	def remove(self)	

Рисунок 3.1 – Діграма класів системи збереження організаційних заходів

3.4.2 Діаграма компонентів

Після того як ми визначилися із усіма обраними технологіями розробки, ми маємо змогу створити діаграму компонентів системи. Загальний її вигляд відобразимо на рисунку нижче (Рисунок 3.2)

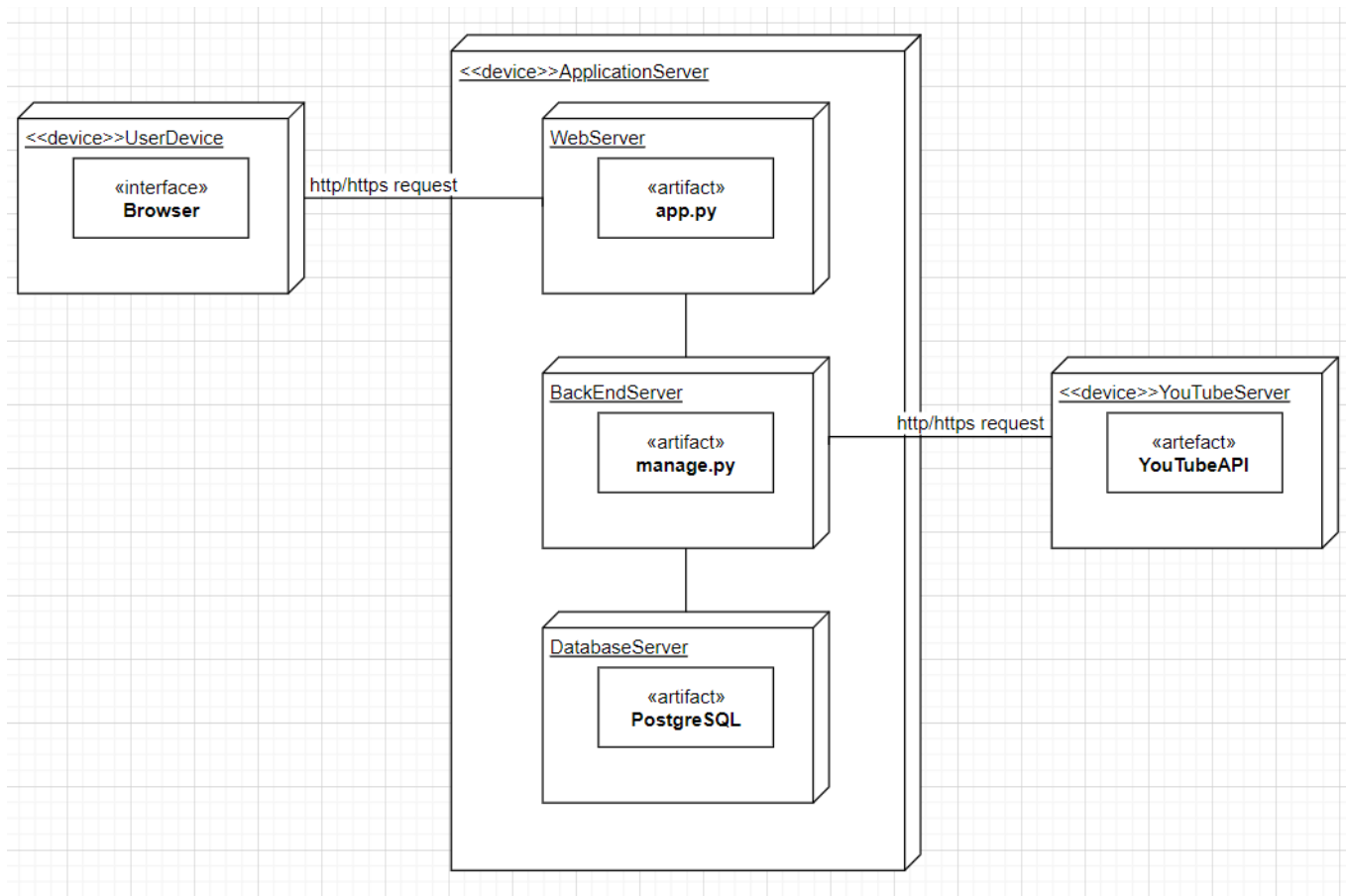


Рисунок 3.2 – Діаграма компонентів системи

3.4.3 Специфікація функцій.

В таблицях 3.1-3.3 описані методи моделей системи, що є класами для взаємодії з базою даних.

Таблиця 3.1 – Опис методів моделі User

Назва	Примітка
Клас: User – клас, який обробляє вхідні дані про користувача.	
public User setFirstName(String firstName)	Установлює ім'я користувача

Продовження таблиці 3.1

public User setLastName(String lastName)	Установлює прізвище користувача
public User setLogin(String login)	Установлює логін користувача
public User setPassword(String password)	Установлює пароль користувача

Таблиця 3.2 – Опис методів моделі Project

Назва	Примітка
Клас: Project – клас, який обробляє вхідні дані для сортування.	
public Project setLabelEvent (String label_event)	Встановлює назву заходу
public Project setAttachFile(String file_path)	Прикріплює файл заходу(doc\docx)
public Project setDescrEvent(String descr_event)	Встановлює опис заходу
public Project search(String pattern)	Виконує пошук серед заходів
public Project sort(int type)	Сортує заходи
public Project setScenario(String text)	Встановлює сценарій заходу
public Project count_like()	Додає до заходу одне «вподобання»
public Project count_dislike()	Додає до заходу одне «невподобання»
public Project count_view()	Додає до заходу один перегляд

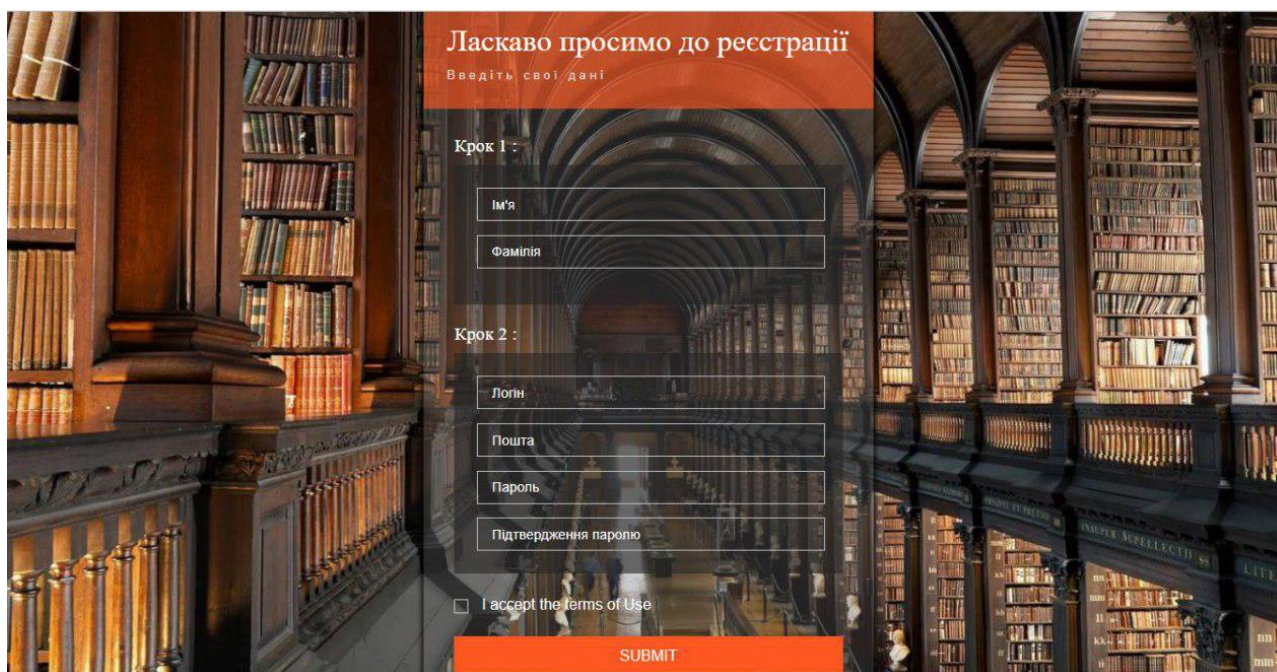
Таблиця 3.3 – Опис методів моделі Content

Висновок до розділу:

Назва	Примітка
Клас: Content – клас, який обробляє вхідні дані для перегляду.	
public Content setAttachFile(String file_url_path)	Прикріплює медіафайл до заходу
public Content setContent_type(int content_type)	Встановлює тип прикріпленого медіафайлу

На даному етапі було розглянуто та проаналізовано такі мови програмування як Python та Ruby, описаний Django фреймворк та середовище розробки в якому був створений додаток, використані API, а також виявлені вимоги до технічного забезпечення, побудова діаграми класів, діаграми компонентів системи та описано специфікацію функцій системи.

- логін;
- пошта;
- пароль



Ласкаво просимо до реєстрації

Введіть свої дані

Крок 1 :

Ім'я

Фамілія

Крок 2 :

Логін

Пошта

Пароль

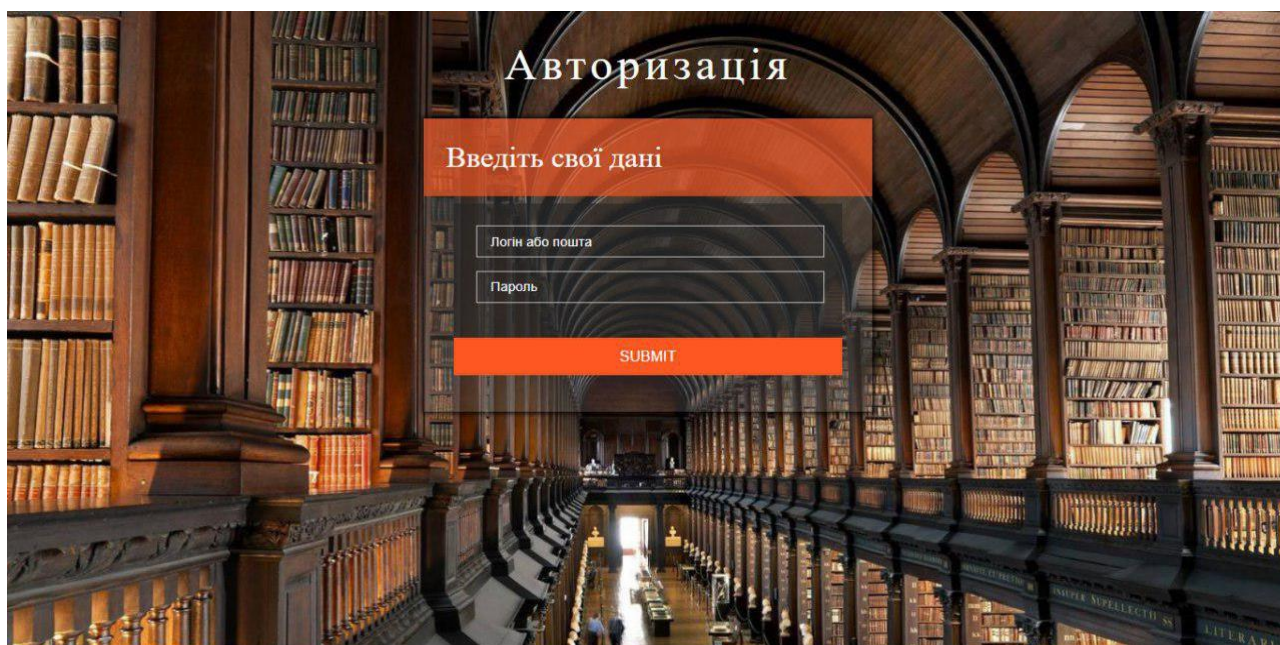
Підтвердження паролю

☐ I accept the terms of Use

SUBMIT

Рисунок 4.1 – Сторінка реєстарції

Після цього зареєстрований користувач має змогу авторизуватися в систему за необхідністю.



Авторизація

Введіть свої дані

Логін або пошта

Пароль

SUBMIT

Рисунок 4.2 – Сторінка авторизації

4.2.2 Сторінка «Мої проекти»

Авторизований користувач має змогу відкрити вкладку «Мої проекти», де він побачить проекти, що були ним завантажені в систему (рисунок 4.3). У даній вкладці користувач має змогу окрім перегляду власних проектів додати новий.

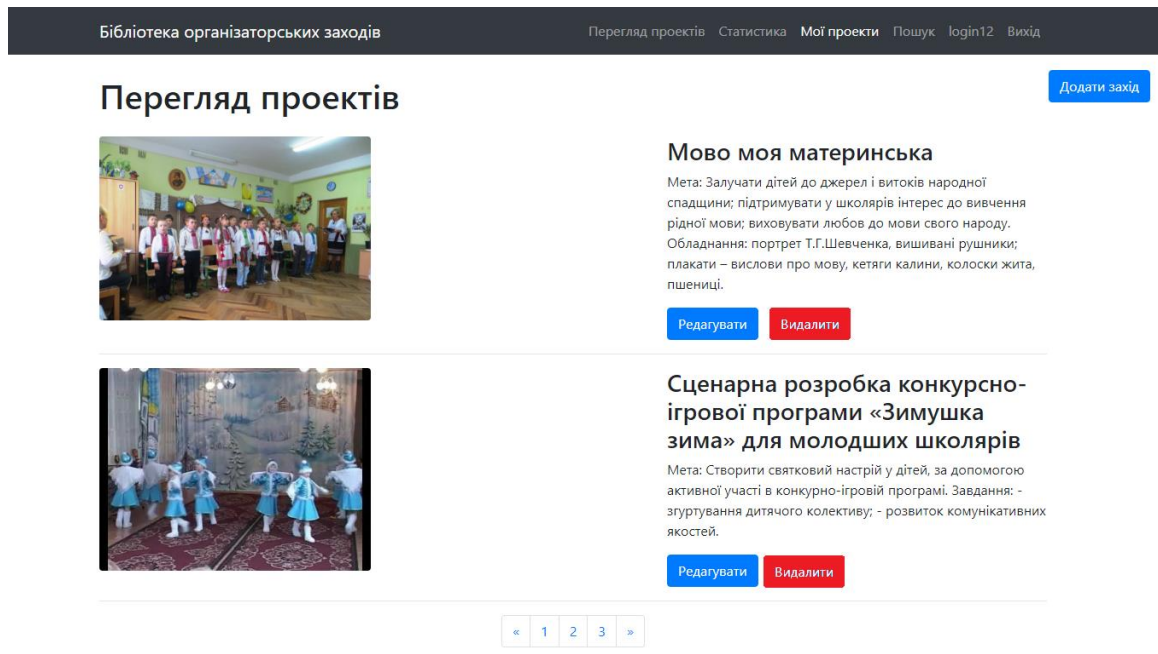


Рисунок 4.3 – Вкладка «Мої проекти»

4.2.3 Сторінка редагування проекту

Авторизований користувач на необхідність може редагувати проект, перейшовши до детального перегляду власного проекту та натиснувши кнопку «Редагувати». Після цих кроків користувач матиме змогу внести зміни у проект (рисунок 4.4).

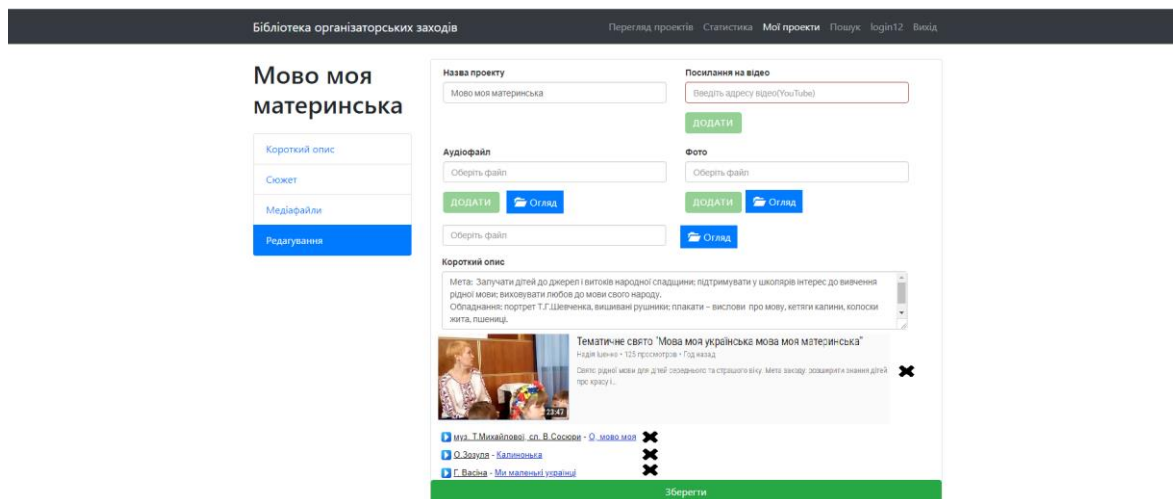


Рисунок 4.4 – Сторінка редагування проекту

4.2.4 Сторінка створення проекту

Авторизований користувач має змогу створити проект, увівши наступні дані:

- назва;
- швидкий опис;
- прикріпити сценарій проекту;

Також опціонально додати:

- посилання на медіафайли;
- прикріпити зображення.

The screenshot shows a web interface for creating a new event. At the top, there is a dark navigation bar with the text 'Бібліотека організаційних заходів' and several links: 'Перегляд проекту', 'Статистика', 'Мої проекти', 'Пошук', 'login12', and 'Вийти'. Below the navigation bar, the main content area is divided into two columns. The left column is titled 'Назва проекту' and contains a text input field. The right column is titled 'Створити новий захід' and contains several sections: 'Назва проекту' with a text input field and a 'Додати' button; 'Посилання на відео' with a text input field and a 'Додати' button; 'Аудіофайл' with a text input field and a 'Додати' button; 'Фото' with a text input field and a 'Додати' button; 'Короткий опис' with a text input field and a 'Додати' button; and 'Обрати файл' with a text input field and a 'Додати' button. At the bottom of the right column, there is a large green button labeled 'Створити'.

Рисунок 4.6 – Сторінка створення нового заходу

4.2.5 Сторінка детального перегляду проекту

Користувач має змогу відкрити на сторінці «Перегляд заходів» один проект й продивитися його більш детально. Для цього йому необхідно обрати кнопку «детальний перегляд». Загальний вигляд сторінки детального перегляду проекту наведений на рисунку 4.7. На цій вкладці користувач також має змогу:

- оцінити проект;
- продивитися сюжет проекту.

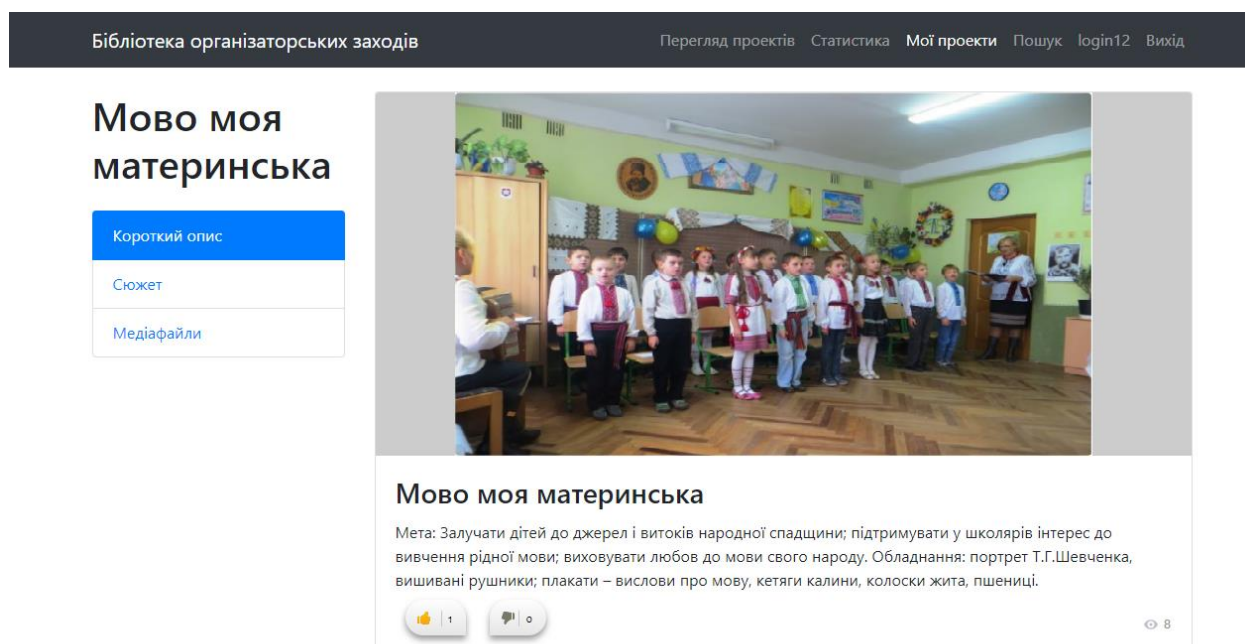


Рисунок 4.7 – Сторінка детального перегляду проекту

4.2.6 Сторінка пошуку

Користувач має змогу скористатися пошуком в системі. Для цього йому необхідно обрати вкладку «Пошук». Загальний вигляд сторінки пошуку проекту наведений на рисунку 4.7.

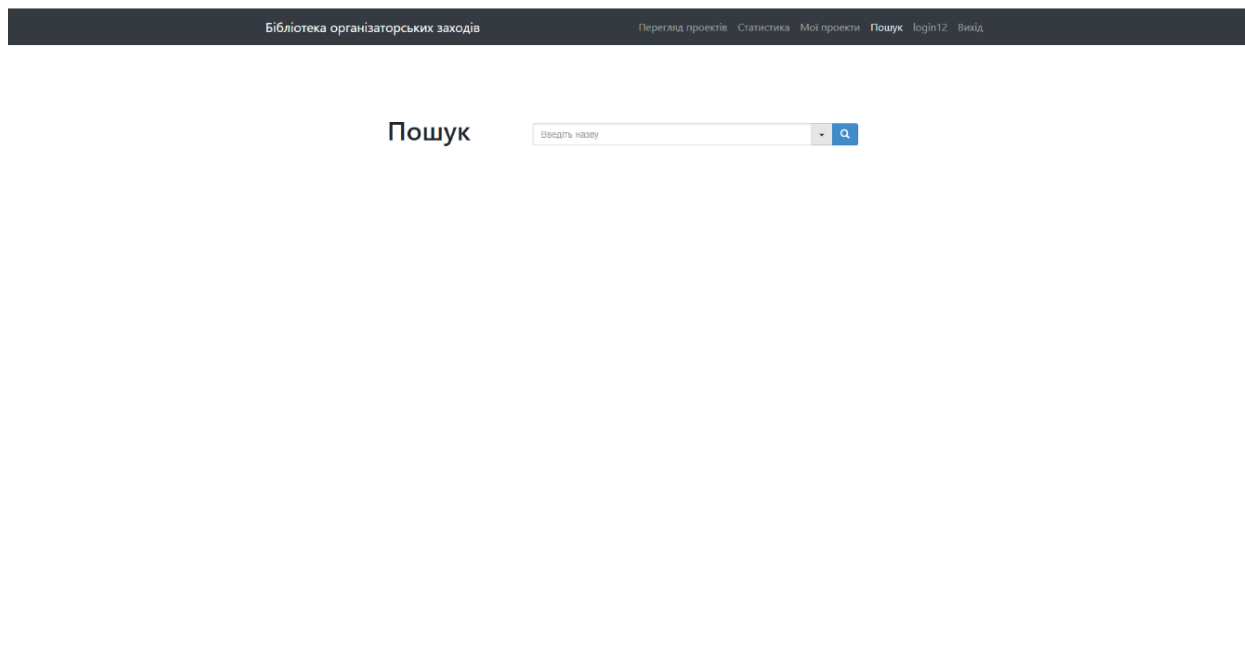


Рисунок 4.8 – Сторінка пошуку заходів

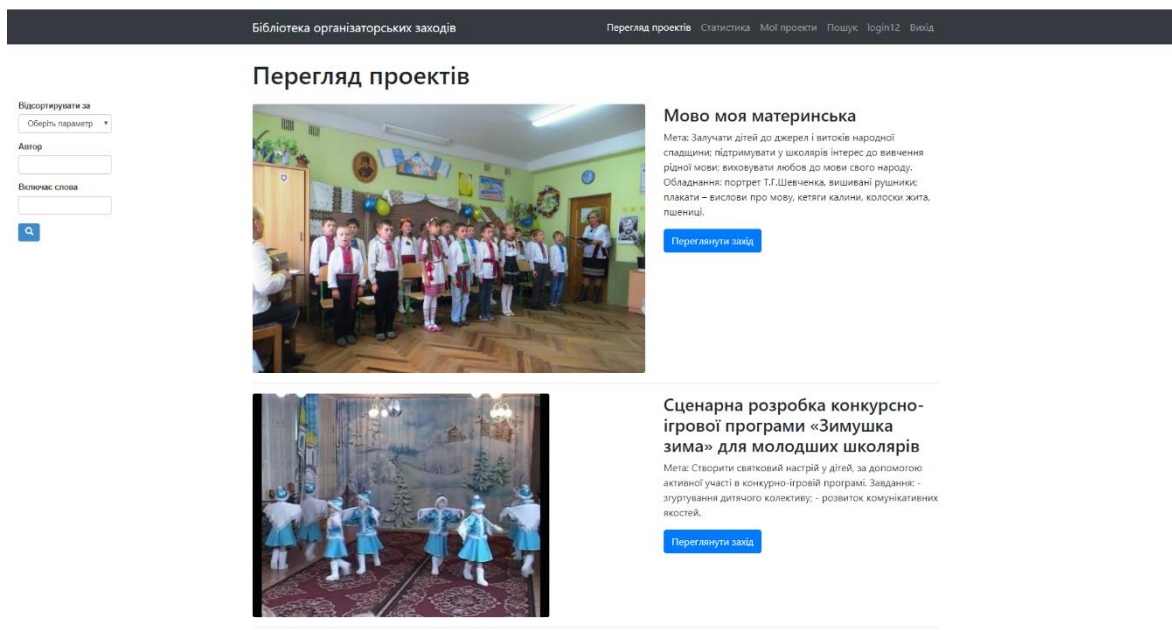


Рисунок 4.9 – Сторінка перегляду заходів

4.3 Випробовування програмного продукту

В цьому підрозділі описано детальне тестування функцій системи для перевірки коректності роботи системи.

4.3.1 Мета випробувань

Метою випробувань є перевірка відповідності функцій створеної системи вимогам технічного завдання.

4.3.2 Загальні положення

Випробування проводяться на основі наступних документів:

- ГОСТ 34.603–92. Інформаційна технологія. Види випробувань автоматизованих систем;
- ГОСТ РД 50-34.698-90. Автоматизовані системи вимог до змісту документів.

4.3.3 Результати випробувань

В процесі тестування були перевірена уся функціональність системи.

Проведемо випробування реалізації основних функцій заявлених у постановці задачі. Наведемо результати їх тестування у вигляді тестових сценаріїв наведених у таблицях 4.1-4.5.

4.3.3.1 Тест додавання проєктів

У зареєстрованого користувача є можливість додати публічний захід. Для цього йому необхідно провести декілька кроків, що описані нижче (таблиця 4.1). В результаті справно працююча система має додати новий захід при відпраці валідних даних. Для перевірки результату скористаємось функцією пошуку в системі.

Таблиця 4.1 – Тест на створення нового проекту

Функція	Створення проекту	
Дія:	Бажаний результат:	Результат тесту: – вдалий – невдалий
Попередні кроки:		
Увійти в систему та відкрити веб-сторінку «Мої проекти»	Веб-сторінка із власними проектами відкрита.	Вдалий

Продовження таблиці 4.1

Кроки тесту:		
Натисніть на кнопку «Створити проект»	Відкрита форма для заповнення даних про проект	Вдалий
Заповніть поля: Назва: «проект» Текст: «тестування нового проекту» Відео: https://youtu.be/F8Rep9ZaFBw	Усі необхідні поля заповнені	Вдалий
Натисніть «Завершити»	Проект завантажений у систему. Відритий детальний перегляд щойно доданого проекту.	Вдалий

4.3.3.2 Тест видалення проектів

У зареєстрованого користувача є можливість видалити свій власний публічний захід. Для цього йому необхідно провести декілька кроків, що описані нижче (таблиця 4.2). В результаті справно працююча система має додати видалити захід з

бази даних проектів. Для перевірки результату скористаємось функцією пошуку в системі.

Таблиця 4.2 – Тест на видалення власного проекту

Функція	Видалення проекту	
Дія:	Бажаний результат:	Результат тесту: – вдалий – невдалий
Попередні кроки:		
Увійти в систему та відкрити веб-сторінку «Мої проекти»	Веб-сторінка із власними проектами відкрита.	Вдалий
Кроки тесту:		
Обрати проект «Тест» та натиснути «Видалити»	Відкрилося вікно із запитанням підтвердження дії	Вдалий
Натисніть «Так»	Проект був видалений із системи	Вдалий

Продовження таблиці 4.2

Післяумова:		
У верхній частині веб-додатку, у текстовому полі пошуку введіть «Тест» та натисніть «Enter»	Виведення веб-сторінки із інформацією про то, що знайдено проектів	Вдалий

4.3.3.3 Тест редагування проектів

У зареєстрованого користувача є можливість редагувати свій власний публічний захід. Для цього йому необхідно провести декілька кроків, що описані нижче (таблиця 4.3). В результаті справно працююча система має заредагувати проект в системі. Для перевірки результату скористаємось функцією детальний перегляд проектів

Таблиця 4.3 – Тест на редагування власного проекту

Функція	Редагування проекту	
Дія:	Бажаний результат	Результат тесту: – вдалий – невдалий
Попередні кроки:		

Бути авторизованим та зайти на сторінку «Мої проекти»	Веб-сторінка із власними проектами відкрита.	Вдалий
Кроки тесту:		
Обрати проект із назвою «Тест» та натиснути «Редагувати»	Відкрилося вікно зміни проекту	Вдалий
У текстовому полі «Назва» напишіть «тест2»	Текстове поле було змінено	Вдалий
Натисніть «Завершити»	Дані по цьому проекту були оновлені. Відритий детальний перегляд щойно відредагованого проекту.	Вдалий

4.3.3.4 Тест детального перегляду проектів

У користувача є можливість детально переглянути будь-який з проектів. Для цього йому необхідно провести декілька кроків, що описані нижче (таблиця 4.4). В результаті справно працююча система має відобразити перегляд проекту у детальному вигляді. Для перевірки необхідно відкрити вікно детального перегляду проектів

Таблиця 4.4 – Тест на детальний перегляд проектів

Функція	Детальний перегляд проектів	
Дія:	Бажаний результат	Результат тесту: – вдалий – невдалий
Попередні кроки:		
Увійти в систему та відкрити веб-сторінку «Мої проекти»	Веб-сторінка із власними проектами відкрита.	Вдалий
Кроки тесту:		
Натиснути кнопку «Перегляд» під бажаним для перегляду проектом	Сторінка детального перегляду була відкрита	Вдалий

4.3.3.5 Тест детального перегляду проектів

У користувача є можливість оцінити будь-який з проектів. Для цього йому необхідно провести декілька кроків, що описані нижче (таблиця 4.5). В результаті справно працююча система має моментально змінити рейтинг проекту у

Таблиця 4.5 – Тест оцінення проекту

Функція	Оцінення проекту	
Дія:	Бажаний результат	Результат тесту: – Вдалий – невдалий
Попередні кроки:		
Увійти в систему та відкрити веб-сторінку «Мої проекти»	Веб-сторінка із власними проектами відкрита.	Вдалий
Натиснути кнопку «Перегляд» під бажаним для перегляду проектом	Сторінка детального перегляду була відкрита	Вдалий
Кроки тесту:		
Натиснути «Like» або «Dislike»		Вдалий

4.3.3.6 Тест авторизації

У неавторизованого користувача є можливість увійти в системи під своїм акаунтом. Для цього йому необхідно провести декілька кроків, що описані нижче (таблиця 4.6). В результаті справно працююча система має відобразити головне вікно додатку у авторизованому режимі.

Таблиця 4.6 – Тест на авторизацію користувача

Функція	Авторизація	
Дія:	Бажаний результат	Результат тесту: – вдалий – невдалий
Кроки тесту:		
Заповніть два текстових поля Пошта: user43245@gmail.com Пароль: user43245	Текстові поля заповнені.	Вдалий

Натисніть «Ввійти в систему».	Відкриття сторінки перегляду проектів	Вдалий
Післяумова:		
Натисніть «Вихід із системи».	Користувач вийшов із системи	Вдалий

4.3.3.7 Тест помилки при неіснуючому користувачі

У неавторизованого користувача є можливість увійти в системи під своїм акаунтом. Для цього йому необхідно провести декілька кроків, що описані нижче (таблиця 4.7). У випадку неспівпадіння логіну із логінами, що є у БД, система має повернути повідомлення про помилку.

Таблиця 4.7 – Тест виведення помилки при неправильному логіні

Функція	Авторизація	
Дія:	Бажаний результат	Результат тесту: – вдалий – невдалий
Попередні кроки:		
Заповніть два текстових поля Пошта: user43245@gmail.com Пароль: user43243	Текстові поля заповнені.	Вдалий
Натисніть «Ввійти в систему».	Відкриття сторінки перегляду проектів	Вдалий
Післяумова:		
Залишити користувача на поточній сторінці задля наступних спроб авторизації	Сторінка авторизації відкрита	Вдалий

4.3.3.8 Тест реєстрації

У користувача є можливість детально переглянути будь-який з проектів. Для

цього йому необхідно провести декілька кроків, що описані нижче (таблиця 4.8). В результаті справно працююча система має створити нового користувача. Для перевірки необхідно заповнити форму авторизації.

Таблиця 4.8 – Тест реєстрації нового користувача

Функція	Реєстрація	
Дія:	Бажаний результат	Результат тесту: – Вдалий – невдалий – заблокований
Кроки тесту:		
Натисніть «Реєстрація» у верхній частині екрану	Відкрита сторінка реєстрації	Вдалий
Заповніть наступні текстові поля Пошта: «janderr97@list.ru» Ім'я: «Євген» Прізвище: «Ілуца» Логін: «user7893» Пароль: «user7893» Підтвердження паролю: «user7893»	Дані успішно введені.	Вдалий
Натисніть «Завершити».	Створено нового користувача.	Вдалий

Продовження таблиці 4.8

Післяумова:		
Заповніть два текстових поля Пошта: user43245@gmail.com Пароль: user43243	Текстові поля заповнені.	Вдалий
Натисніть «Ввійти в систему».	Відкриття сторінки перегляду проектів	Вдалий

4.3.3.9 Тест виведення помилки при невірному паролі

У неавторизованого користувача є можливість зареєструватися в систему. Для цього йому необхідно провести декілька кроків, що описані нижче (таблиця 4.9). У випадку неспівпадіння паролю із паролем даного логіну, що є у БД, система має повернути повідомлення про помилку.

Таблиця 4.9 – Тест виведення помилки при невірному паролі

Функція	Реєстрація	
Дія:	Бажаний результат	Результат тесту: – вдалих – невдалих
Кроки тесту:		
Натисніть «Реєстрація» у верхній частині екрану	Відкрита сторінка реєстрації	Вдалих
Заповніть наступні текстові поля Пошта: «janderr97@list.ru» Ім'я: «Євген» Прізвище: «Ілуща» Логін: «user7893» Пароль: «user7» Підтвердження паролю: «user7»	Дані успішно введені.	Вдалих
Натисніть «Завершити».	Виведення помилки про недостатню довжину паролю	Вдалих
Післяумова:		
Сторінка реєстрації досі відкрита для наступних спроб реєстрації	Сторінка реєстрації досі відкрита	Вдалих

4.3.3.10 Тест виведення помилки при невірній пошті

У неавторизованого користувача є можливість зареєструватися в систему.

Для цього йому необхідно провести декілька кроків, що описані нижче (таблиця 4.10).

У випадку невалідності пошти, система має повернути повідомлення про помилку.

Таблиця 4.10 – Тест виведення помилки при невірному паролі

Функція	Реєстрація	
Дія:	Бажаний результат	Результат тесту: – вдалих – невдалих
Кроки тесту:		
Натисніть «Реєстрація» у верхній частині екрану	Відкрита сторінка реєстрації	Вдалих
Заповніть наступні текстові поля Пошта: «janderr97@list» Ім'я: «Євген»	Дані успішно введені.	Вдалих

Прізвище: «Ілуца» Логін: «user7898» Пароль: «user7898» Підтвердження паролю: «user7898»		
Натисніть «Завершити».	Виведення помилки про невалідність пошти	Вдалий
Післяумова:		
Сторінка реєстрації досі відкрита для наступних спроб реєстрації	Сторінка реєстрації досі відкрита	Вдалий

4.3.3.11 Тест виведення помилки при займаному логіні

У неавторизованого користувача є можливість зареєструватися в систему під Для цього йому необхідно провести декілька кроків, що описані нижче (таблиця 4.11). У випадку зайнятого логіну, система має повернути повідомлення про помилку.

Таблиця 4.11 – Тест виведення помилки при займаному логіні

Функція	Реєстрація		
Дія:	Бажаний результат	Результат тесту: – вдалий – невдалий	
Кроки тесту:			
Натисніть «Реєстрація» у верхній частині екрану	Відкрита сторінка реєстрації	Вдалий	
Заповніть наступні текстові поля Пошта: «janderr97@list.ru» Ім'я: «Євген» Прізвище: «Ілуца» Логін: «user7897» Пароль: «user7897» Підтвердження паролю: «user7897»	Дані успішно введені.	Вдалий	

Натисніть «Завершити».	Виведення помилки про те, що логін вже зайнятий іншим користувачем	Вдалий
Післяумова:		
Сторінка реєстрації досі відкрита для наступних спроб реєстрації	Сторінка реєстрації досі відкрита	Вдалий

4.3.3.12 Тест виведення помилки при неспівпадінні паролів

У неавторизованого користувача є можливість зареєструватися в систему під Для цього йому необхідно провести декілька кроків, що описані нижче (таблиця 4.12). У випадку неспівпадіння паролів при реєстрації система має вивести повідомлення про помилку.

Таблиця 4.12 – Тест виведення помилки при неспівпадінні паролів

Функція	Реєстрація		
Дія:	Бажаний результат	Результат тесту: – вдалий – невдалий	

Продовження таблиці 4.12

Кроки тесту:		
Натисніть «Реєстрація» у верхній частині екрану	Відкрита сторінка реєстрації	Вдалий
Заповніть наступні текстові поля Пошта: «janderr97@list.ru» Ім'я: «Євген» Прізвище: «Ілуца» Логін: «user7899» Пароль: «user7899» Підтвердження паролю: «user7898»	Дані успішно введені.	Вдалий
Натисніть «Завершити».	Виведення помилки про те, що паролі не співпадають	Вдалий

Післяумова:		
Сторінка реєстрації досі відкрита для наступних спроб реєстрації	Сторінка реєстрації досі відкрита	Вдалий

4.3.3.13 Тест пошуку в системі

У користувача є можливість виконати пошук в системі за допомогою текстового поля. Для цього йому необхідно провести декілька кроків, що описані нижче (таблиця 4.13). Система має повернути проекти, що задовільняють вказаному патерну.

Таблиця 4.13 – Тест пошуку в системі

Дія:	Бажаний результат	Результат тесту: – вдалий – невдалий
Попередні кроки:		
Заповніть текстове поле «Пошук»: Ізмаїл 2019	Текстове поле заповнено.	Вдалий
Натисніть «Enter».	Відкрит детальний перегляд проекту «День міста Ізмаїл 2019»	Вдалий

4.3.3.14 Тест виводу сторінки про відсутність проектів у пошуку

У користувача є можливість виконати пошук в системі за допомогою текстового поля. Для цього йому необхідно провести декілька кроків, що описані нижче (таблиця 4.14). У випадку якщо система не знайшла проекти, що задовільняють вказаному патерну, необхідно вивести вікно про відсутність проектів за вказаним патерном.

Таблиця 4.14 – Тест виводу сторінки про відсутність проектів у пошуку

Функція	Пошук	
Дія:	Бажаний результат	Результат тесту: – вдалий – невдалий

Кроки тесту:		
Заповніть текстове поле «Пошук»: Тестовий запит	Текстове поле заповнено.	Вдалий
Натисніть «Enter».	Відкрито сторінку про відсутність проектів за вказаною назвою	Вдалий

4.3.3.15 Тест сортування за автором

У користувача є можливість сортувати проекти за автором в систему.

Для цього йому необхідно провести декілька кроків, що описані нижче (таблиця 4.15).

Система має повернути проекти відсортовані за автором проекту.

Таблиця 4.15 – Тест сортування за автором

Функція	Сортування	
Дія:	Бажаний результат	Результат тесту: – вдалий – невдалий
Попередні кроки:		
Відкрити сторінку перегляду проектів	Сторінка перегляду проектів відкрита	Вдалий
Кроки тесту:		
Оберіть тип сортування «Автор»	Тип сортування «Автор» був обраний	Вдалий
Натисніть «ОК»	Відкрита сторінка із відсортованими проектами за алфавітом за допомогою поля «автор»	Вдалий

4.3.3.16 Тест сортування проектів за їх назвою

У користувача є можливість сортувати проекти за назвою проекту в систему.

Для цього йому необхідно провести декілька кроків, що описані нижче (таблиця 4.16).

Система має повернути проекти відсортовані за назвою проекту.

Таблиця 4.16 – Тест сортування проектів за їх назвою

Функція	Сортування
---------	------------

Дія:	Бажаний результат	Результат тесту: – вдалий – невдалий
Попередні кроки:		
Відкрити сторінку перегляду проектів	Сторінка перегляду проектів відкрита	Вдалий
Кроки тесту:		
Оберіть тип сортування «Назва»	Тип сортування «Назва» був обраний	Вдалий
Натисніть «ОК»	Відкрита сторінка із відсортованими проектами за алфавітом за допомогою поля «назва»	Вдалий

4.3.3.17 Тест сортування за значенням «найновіші»

У користувача є можливість сортувати проекти за датою завантаження проекту в систему. Для цього йому необхідно провести декілька кроків, що описані нижче (таблиця 4.17). Система має повернути проекти відсортовані за датою завантаження.

Таблиця 4.17 – Тест сортування за значенням «найновіші»

Функція	Сортування	
Дія:	Бажаний результат	Результат тесту: – вдалий – невдалий
Попередні кроки:		
Відкрити сторінку перегляду проектів	Сторінка перегляду проектів відкрита	Вдалий
Кроки тесту:		
Оберіть тип сортування «найновіші»	Тип сортування «найновіші» був обраний	Вдалий
Натисніть «ОК»	Відкрита сторінка із відсортованими проектами за датою публікації	Вдалий

4.3.3.18 Тест сортування за значенням «найбільш популярні»

У користувача є можливість сортувати проекти за рейтингом популярності проекту в систему. Для цього йому необхідно провести декілька кроків, що описані нижче (таблиця 4.18). Система має повернути проекти відсортовані за рейтингом популярності проекту.

Таблиця 4.18 – Тест сортування за значенням «найпопулярні»

Функція	Сортування	
Дія:	Бажаний результат	Результат тесту: – вдалих – невдалих
Попередні кроки:		
Відкрити сторінку перегляду проектів	Сторінка перегляду проектів відкрита	Вдалих
Кроки тесту:		
Оберіть тип сортування «найпопулярні»	Тип сортування «найпопулярні» був обраний	Вдалих
Натисніть «ОК»	Відкрита сторінка із відсортованими проектами за найшвидшим ростом рейтингу за останній час	Вдалих

4.3.3.19 Тест сортування за значенням «топ рейтинг»

У користувача є можливість сортувати проекти за рейтингом проекту в систему. Для цього йому необхідно провести декілька кроків, що описані нижче (таблиця 4.19). Система має повернути проекти відсортовані за рейтингом проекту.

Таблиця 4.19 – Тест сортування за значенням «топ рейтинг»

Функція	Сортування	
Дія:	Бажаний результат	Результат тесту: – вдалих – невдалих

Попередні кроки:		
Відкрити сторінку перегляду проектів	Сторінка перегляду проектів відкрита	Вдалий
Кроки тесту:		
Оберіть тип сортування «топ рейтинг»	Тип сортування «топ рейтинг» був обраний	Вдалий
Натисніть «ОК»	Відкрита сторінка із відсортованими проектами із найвищим рейтингом за увесь час	Вдалий

Висновки до розділу:

У даному розділі було розглянуто керівництво користувача, в якому були детально описані увесь функціонал системи, а також її тестування, яке система успішно пройшла, набравши максимальну можливу кількість балів.

5 РОЗРОБКА СТАРТАП-ПРОЕКТУ

5.1 Вступ

Пояснювальна записка та програмне забезпечення, що було створені під час виконання цієї магістерської дисертації можуть мати комерційну цінність наукових, освітніх або ж інших державних організаціях. Системи, що націлені саме на архівування публічних заходів на даний момент не існує.

5.2 Опис ідеї стартап-проекту

Ідея стартап-проекту описана у таблиці 5.1.

Таблиця 5.1 – Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Система управління контентом публічних заходів	1. Створення платформи архівування публічних заходів	Створення моделі для надання широкодоступного та зручного доступу до веб-застосунку архівування публічних заходів

	2. Пошук та найм спеціалістів з підтримання веб-додатку у належному стані та розширення, за необхідністю, функціоналу	Впевненість у якості роботи платформи.
--	---	--

Продовження таблиці 5.1

Зміст ідеї	Напрямки застосування	Вигоди для користувача
	3. Пошук та підготовка спеціалістів з питань технічної підтримки	Наявність кваліфікаційних співробітників у питаннях технічної підтримки.

Аналіз сильних, слабких та нейтральних сторін створеної платформи наведено у таблиці 5.2. Перевагами системи є низька (відповідно ринку) вартість користування, оптимізований дизайн під архівування публічних заходів, наявність оцінювання заходів, статистики та групування проектів.

Як аналоги було розглянуто тільки хмарні сховища, адже інші варіанти сильно поступаються першим. Нагадуємо, що аналоги розглядаються в задачі архівування публічних заходів.

Таблиця 5.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

		(потенційні) товари/концепції конкурентів			W (слабка сторона)	N (нейтра льна сторон а)	S (сильна сторона)
		Мій проект	Google	Dropbox			
1	Вартість експлуатації	200 грн за місяць	Від 5,00 доларів США/мі сяць за користу вача	Від 10,00 доларів США/мі сяць за користув ача	-	-	+
2	Кросплатформенність	+	+	+	-	+	-
3	Наявність технічної підтримки на українській, російській та англійській мові	+	-	-	-	-	+
4	Зручність користування графічним інтерфейсом	Висока	Низька	Низька	-	-	+

Таблиця 5.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

		(потенційні) товари/концепції конкурентів			W (слабка сторона)	N (нейтра льна сторон а)	S (сильна сторона)
		Мій проект	Google	Dropbox			
5	Наявність можливості роботи з різних пристроїв	+	+	-	-	-	+
6	Стійкість до відмов	Висока	Висока	Висока	-	+	-
7	Наявність системи резервування	+	+	+	-	+	-
8	Ступінь розширюваності	Висока	Середня	Низька	-	-	+
9	Робота з великою кількістю сервісів	-	+	+	+	-	-
11	Наявність ретингу проектів	+	-	-	-	-	+

Таблиця 5.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

		(потенційні) товари/концепції конкурентів			W (слабка сторона)	N (нейтра льна сторон а)	S (сильна сторона)
		Мій проект	Google	Dropbox			
12	Швидкість і точність пошуку проектів	Низька	Висока	Середня	+	-	-
13	Наявність статистики проектів	+	-	-	-	-	+
14	Наявність групування проектів	+	-	-	-	-	+
15	Миттєвий перегляд медіа файлів	+	-	-	-	-	+
15	Налаштування прав доступу	+	+	+	-	+	-

Таблиця 5.3. Технологічна здійсненність ідеї проекту

№ п/ п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1.	Створення платформи архівування публічних заходів	Технологія 1 (технологія виготовлення товару, надання послуги)	Чи вони наявні, або ж необхідно їх розробити/додати?	Чи вони доступні авторам проекту?
2.	Створення веб-додатку	Django Python	Наявна	Доступні
3.	Програмування логіки	Python	Наявна	Доступна
4.	Підтримка бази даних проектів	PostgreSQL	Наявна	Доступна

Обрана технологія реалізації ідеї проекту: з огляду на те що всі технології для реалізації ідей доступні, то ми можемо реалізувати всі заплановані ідеї.

Таблиця 5.1. Попередня характеристика потенційного ринку стартап-проекту

№ п/ п	Показники стану ринку (найменування)	Характеристика
--------------	--------------------------------------	----------------

1.	Кількість головних гравців, од	2 (Dropbox, Google Drive)
2.	Загальний обсяг продаж, грн/ум.од	200 грн/міс
3.	Динаміка ринку (якісна оцінка)	Зростає
4.	Наявність обмежень для входу (вказати характер обмежень)	Юридичні обмеження

Продовження таблиця 5.4

5.	Специфічні вимоги до стандартизації та сертифікації	Згідно регламентам, що описують вимоги до створення програмного забезпечення.
6.	Середня норма рентабельності в галузі (або по ринку), %	80 %

Ринок є привабливим, оскільки є потреба у системі архівування публічних заходів

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1.	Архівування публічних заходів	Організатори публічних заходів	Технології розробки програмного забезпечення.	Безпека особистих даних, інтуїтивна зрозумілість інтерфейсу, наявність необхідного функціоналу, зазначеного вище. Необхідність

				наявності технічної підтримки.
--	--	--	--	--------------------------------

Аналіз факторів загроз та можливостей допоможе зекономити гроші та час у майбутньому, а також допоможе сформулювати задачі для подальшого розвитку системи. У таблиці 4.6 визначені фактори загроз.

Таблиця 5.2. Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1.	Некваліфікованість користувачів	Таргетингова група користувачів – організатори публічних заходів. Є така ймовірність, що вони не будуть спроможні використовувати увесь функціонал платформи за необхідністю.	Окрім технічної підтримки – створення документації користування платформою
2.	Нові технології розробки програмного забезпечення	Новий функціонал необхідний для користувачів	Моніторинг відгуків та пропозицій від користувачів

У таблиці 4.7 визначені фактори можливостей.

Таблиця 4.3. Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1.	Зацікавленість організаторів публічних заходів	Створення платформи, що буде зручною для архівування та збереження публічних заходів.	Взаємовигідна співпраця з іншими компаніями.

Продовження таблиця 5.6

2.	Відсутність аналогів	В Україні не має аналогів такої системи, яка б була сфокусовано на обраній тематиці «Публічні заходи»	Можливість стати монополістом в даній галузі
----	----------------------	---	--

У таблиці 5.7 описано ступеневий аналіз конкуренції на ринку.

Таблиця 5.7. Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Вказати тип конкуренції Олігополія	Існують схожі програмні засоби, проте вони не є суто тематичними, а тому не є такими зручними	Інтегрувати найкращі функціональні вимоги аналогів, впровадити нові, що користуються попитом у організаторів публічних заходів

2. За галузевою ознакою Внутрішньогалузева	Боротьба серед аналогів програм за клієнта	Необхідно створити кращий товар ніж у конкурентів
1. За характером конкурентних переваг Нецінова	Програмне забезпечення буде відрізнитися від звичайних хмарних сховищ.	Необхідно вивчати ринок та за допомогою створення фінансової моделі оптимізувати прибуток.

Продовження таблиця 5.7

4. За інтенсивністю Не марочна	Фірма ще не набула достатньої популярності	Необхідність у створенні дійсно якісного продукту для завоювання довіри клієнтів
-----------------------------------	---	---

У таблиці 5.8 представлено Аналіз конкуренції в галузі за М. Портером.

Таблиця 5.8. Аналіз конкуренції в галузі за М. Портером

	Прямі конкуренти в галузі	Потенційні конкуренти	Постачал ьники	Клієнти	Товари- замінники
Складові аналізу	Навести перелік прямих конкурентів	Визначити бар'єри входження в ринок	Визначити фактори сили постачаль ників	Визначити фактори сили споживачі в	Фактори загроз з боку замінників
	Dropbox, Google Drive	Популярність та довіра	-	Гроші	Часткова заміна

		серед користувачів			
Висновки:	Визначити інтенсивність конкурентної боротьби з боку прямих конкурентів	- чи є можливості входу в ринок? - чи є потенційні конкуренти?	Чи постачальники диктують умови роботи на ринку? Які?	Чи клієнти диктують умови роботи на ринку? Які?	Обмеження для роботи на ринку через товари замітники

Продовження таблиця 5.8

Дуже інтенсивна	Є потенційні конкуренти, але вони не повністю задовільняють потреби користувачів. Для входу на ринок необхідно пройти багато бюрократичних процесів.	-	Клієнти вимагають зручність та швидкість завантаження публічних заходів	Товари замітники частково перекривають розроблений програмний продукт, але не дозволяють покрити усі необхідні варіанти використання в сфері збереження публічних заходів
-----------------	--	---	---	---

У таблиці 5.9 наведено обґрунтування факторів конкурентоспроможності.

Таблиця 5.9. Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1.	Ціна	Деякі аналоги є невиправдано дорогими.
2.	Легкість у користуванні	Інтерфейс програмного продукту інтуїтивно зрозумілий, а також увесь функціонал є строго тематичним для збереження публічних заходів

Продовження таблиці 5.9

3.	Можливість подальшого розвитку та завоювання нових сфер	В майбутньому можна використовувати створену платформу в різних державних або приватних організаціях
4.	Задоволення потреб користувачів.	Функціональні вимоги постійно розвиваються, у разі низької кількості користувачів, можливе несвоячасне виявлення необхідності нового функціоналу.

У таблиці 5.10 наведено порівняльний аналіз сильних та слабких сторін проекту.

Таблиця 5.10. Порівняльний аналіз сильних та слабких сторін проекту

п/п	Фактор конкурентоспроможності	Рейтинг товарів-конкурентів у порівнянні з моєю системою						
		-3	-2	-1	0	+1	+2	+3
1.	Ціна						+	

2.	Легкість у користуванні						+	
3.	Можливість подальшого розвитку та завоювання нових сфер					+		
4.	Задоволення потреб користувачів							+

У таблиці 5.11 наведено SWOT- аналіз стартап-проекту.

Таблиця 5.11. SWOT- аналіз стартап-проекту

Сильні сторони: зручність збереження публічних заходів, рейтинг проектів, групування заходів, наявність висококваліфікованих технічної підтримки	Слабкі сторони: відсутність інтеграції веб-сервісу із іншими програмами забезпеченнями, не дуже якісний пошук, в порівнянні із іншими аналогами
Можливості: зацікавлення державних або приватних організаціях в цьому веб-додатку, а також наступної рекламної інтеграції в ньому.	Загрози: недостатня зацікавленість таргетингової групи.

Таблиця 5.12. Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний	Ймовірність отримання ресурсів	Строки реалізації
----------	------------------------------	-----------------------------------	-------------------

	комплекс заходів) ринкової поведінки		
1.	Стратегія спеціалізації	90%	1,5 роки
2.	Стратегія диференціації	50%	2 роки
3.	Позиціонування за співвідношенням "ціна - якість"	98%	1 рік
4.	Стратегія зайняття конкурентної ніші	50%	1 рік

Найкращим варіантом буде стратегія позиціонування за співвідношенням оскільки імовірність отримання ресурсів та строки реалізації є найбільш оптимальними.

У таблиці 5.13 наведено вибір цільових груп потенційних споживачів.

Таблиця 5.13. Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1.	Організатори публічних заходів	Готові	100%	Є декілька конкурентів	Не просто
Які цільові групи обрано: організатори публічних заходів; компанії, що організовують заходи на замовлення.					

У таблиці 5.14 визначено базову стратегію розвитку.

Таблиця 5.14. Визначення базової стратегії розвитку

№п /п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні і позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1.	Зосередження на одному сегменті	Стратегія концентра-ного маркетингу	Відсутність великих ресурсів	Стратегія спеціалізації

У таблиці 5.15 визначено базову стратегію конкурентної поведінки.

Таблиця 5.15. Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента?	Стратегія конкурентної поведінки*
1.	Ні	Нові та існуючі	Так	Стратегія виклику лідера

У таблиці 5.16 визначено стратегію позиціонування.

Таблиця 5.16. Визначення стратегії позиціонування

№п /п	Вимоги до товару	Базова стратегія розвитку	Ключові конкурентоспроможні	Вибір асоціацій, які мають сформувати комплексну
----------	---------------------	---------------------------------	--------------------------------	---

	цільової аудиторії		позиції власного стартап-проекту	позицію власного проекту (три ключових)
1.	Зручність та легкість завантаження заходів	Позиціювання за співвідношенням "ціна - якість"	Оптимальна ціна та якість проекту; Задоволення потребам організаторів публічних заходів	Оптимальна ціна, задоволення потребам, якість та надійність

5.3 Розроблення маркетингової програми стартап-проекту

У таблиці 5.17 визначено ключові переваги концепції потенційного товару

Таблиця 5.17. Ключові переваги концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1.	Архівування заходів у зручному вигляді	Зручний веб- ресурс для збереження публічних заходів в суто тематичному веб-додатку, із можливістю групування та функціоналом рейтингу цих заходів.	На відміну від більшості аналогів, у даному веб-застосунку буде функціонал рейтинг, групування та миттєвого перегляду медіа файлів в самій системі.

У таблиці 5.18 описано три рівні моделі товару.

Таблиця 5.18. Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові
I. Товар за задумом	Програмний продукт для збереження публічних заходів
	Властивості/характеристики
	<ol style="list-style-type: none"> 1. Особистий кабінет. 2. Система створення та редагування заходів. 3. Перегляд заходу. 4. Пошук, групування. 5. Система рейтингу заходів
	До продажу: гарантія надійності та якості
	Після продажу: підтримка актуальності
За рахунок чого потенційний товар буде захищено від копіювання: захист інтелектуальної власності.	

Необхідною складовою є аналіз ціни товару, що дозволяє планувати кількість ресурсів для розробки даного продукту.

У таблиці 5.19 описано формування системи збуту.

Таблиця 5.19. Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
	Приватні організатори публічних заходів; компанії, що спеціалізуються на організації заходів	Надавання якісних функціональних можливостей	Перший рівень	Напрямку

У таблиці 5.20 наведено концепцію маркетингових комунікацій.

Таблиця 5.20. Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1.	Орієнтовані на якість.	Маркетинг.	Якість, надійність, актуальність, адекватність.	Показати надійність та якість продукту.	Продукт є найнадійнішим серед аналогів та користується довірою.

Висновки до розділу:

У даном розділі система була розглянута у якості потенційно комерційного програмного продукту: описана ідея стартап-проекту, а також було розроблено маркетингову стратегію стартап-проекту. Підсумовуючи, можна сказати, що система може бути прибутковою, адже в даній програмі є великий попит користувачів, а також хороші результати у порівняльній аналіз сильних та слабких сторін проекту у порівнянні із існуючими рішеннями.

ВИСНОВКИ

В результаті написання магістерської дисертації була створена система управління контентом публічних заходів, що має ряд переваг в порівнянні з аналогами.

Було проаналізовано процеси збереження та архівування публічних заходів та розглянуто вже існуючі вирішення. На основі їх аналізу було виявлено основні недоліки таких систем та формування нових функціональних вимог для нашого програмного продукту.

Перед реалізацією веб-додатку було створено схему структурної діяльності та формування таблиці усіх функціональних вимог, виявлено вхідні та вихідні дані, а також спроектовано цілісну й нормалізовану базу даних системи.

Для розробки програмного забезпечення було попередньо розглянуто існуючі засоби розробки. Результатом такого аналізу є надання переваги платформі Python, СКБД – PostgreSQL. Були описані функції класів програмного забезпечення, побудовано діаграму класів та компонентів системи.

При розробці користування користувача було надано детальний опис кожної функціональної вимоги, наведено приклад її використання, а також її подальше тестування в залежності від різних вхідних даних.

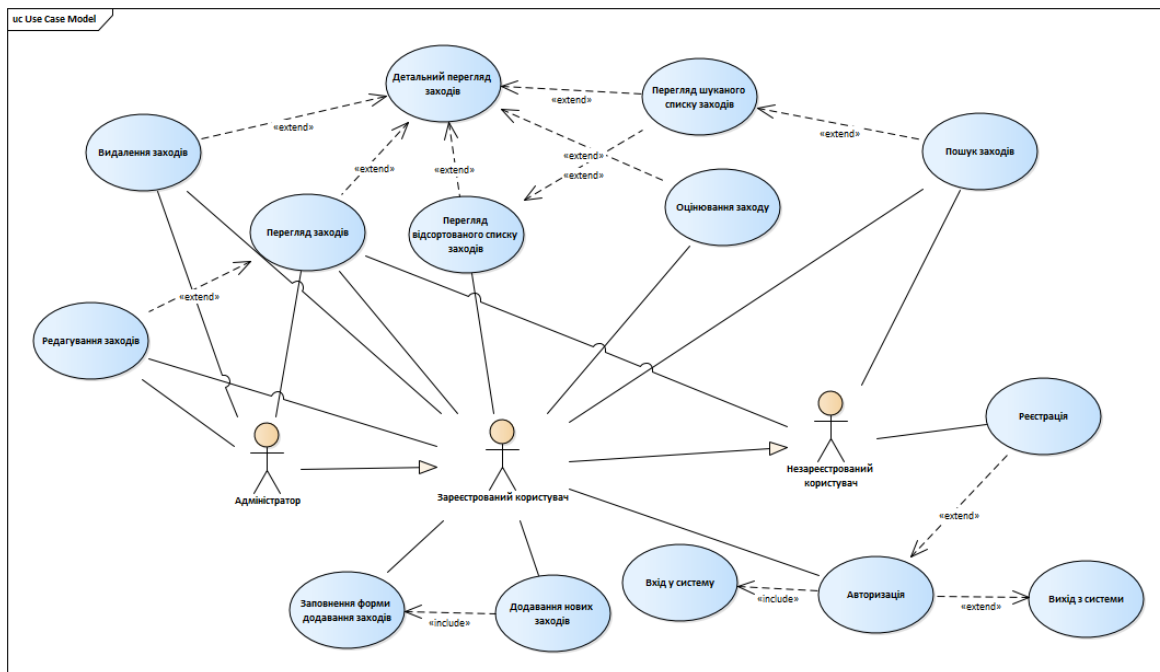
Для майбутнього розвитку проекту було вирішено націлитися на організаторів публічних заходів. Була доведена потреба у розробці такого програмного забезпечення. Впровадження проекту є перспективним, оскільки аналоги не забезпечують покриття усіх функціональних вимог користувачів.

ПЕРЕЛІК ПОСИЛАНЬ

1. Google Cloud Storage Enterprise Data Connector [Електронний ресурс]. – 2016. – Режим доступу до ресурсу: <https://www.treasuredata.com/data-integrations-cloud-storage/google-cloud-storage-data-connector>.
2. A brief overview of Dropbox [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: http://ask-leo.com/a_brief_overview_of_dropbox.html.
3. Overview of all my folders/files in Dropbox [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: <https://www.dropboxforum.com/t5/Files-folders/Overview-of-all-my-folders-files-in-Dropbox/td-p/365510>.
4. Messaging App Development [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: <https://yalantis.com/blog/messaging-apps-development-telegram-whatsapp-others-work>.
5. Creative Blogging: Your First Steps to a Successful Blog [Електронний ресурс]. – 2016. – Режим доступу до ресурсу: <https://books.google.com.ua/books?id=ronlWbUptiwC&pg=PA314&lpg=PA314&dq=vk+for+blogging&source=bl&ots=UrSM8LjDc5&sig=ACfU3U3F21FKb-oCSfheQnu1I30HMbE-kA&hl=ru&sa=X&ved=2ahUKEwjp7-aok7zmAhWzxMQBHfMNAPY4ChDoATABegQIChAB#v=onepage&q=vk%20for%20blogging&f=false>
6. Python vs. Other Programming Languages [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: <https://stxnext.com/python-vs-other-programming-languages>.
7. What is Ruby? - Computer Hope [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: <https://www.computerhope.com/jargon/r/ruby.htm>.
8. Symbol in Ruby - rubycademy - Medium [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: <https://medium.com/rubycademy/symbol-in-ruby-daca5abd4ab2>.
9. Ruby vs. Python: What's the Difference [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: <https://learn.onemonth.com/ruby-vs-python>.

10. Python vs. Ruby: A Comparison of Differences and Similarities [Электронный ресурс]. – 2019. – Режим доступа до ресурсу:
<https://stxnnext.com/blog/2019/09/30/python-vs-ruby-comparison/>.
11. PostgreSQL - Overview [Электронный ресурс]. – 2017. – Режим доступа до ресурсу: https://www.tutorialspoint.com/postgresql/postgresql_overview.htm.
12. REST — A Simple Explanation [Электронный ресурс]. – 2015. – Режим доступа до ресурсу: <https://medium.com/extend/what-is-rest-a-simple-explanation-for-beginners-part-1-introduction-b4a072f8740f>.
13. Python Quickstart | YouTube Data API | Google Developers [Электронный ресурс]. – 2019. – Режим доступа до ресурсу:
<https://developers.google.com/youtube/v3/quickstart/python>.
14. Documentation: 9.5: Advanced Features - PostgreSQL [Электронный ресурс]. – 2016. – Режим доступа до ресурсу:
<https://www.postgresql.org/docs/9.0/tutorial-advanced.html>.
15. MySQL Security: Overview of MySQL security features - PSCE [Электронный ресурс]. – 2015. – Режим доступа до ресурсу:
<https://www.psce.com/en/blog/2013/02/06/mysql-security-overview-of-mysql-security-features/>.
16. MySQL 8 Security Tips - Database Journal [Электронный ресурс]. – 2014. – Режим доступа до ресурсу:
<https://www.databasejournal.com/features/mysql/mysql-8-security-tips.html>.
17. MySQL Enterprise Encryption Function Reference [Электронный ресурс]. – 2018. – Режим доступа до ресурсу:
<https://dev.mysql.com/doc/refman/8.0/en/enterprise-encryption-function-reference.html>.

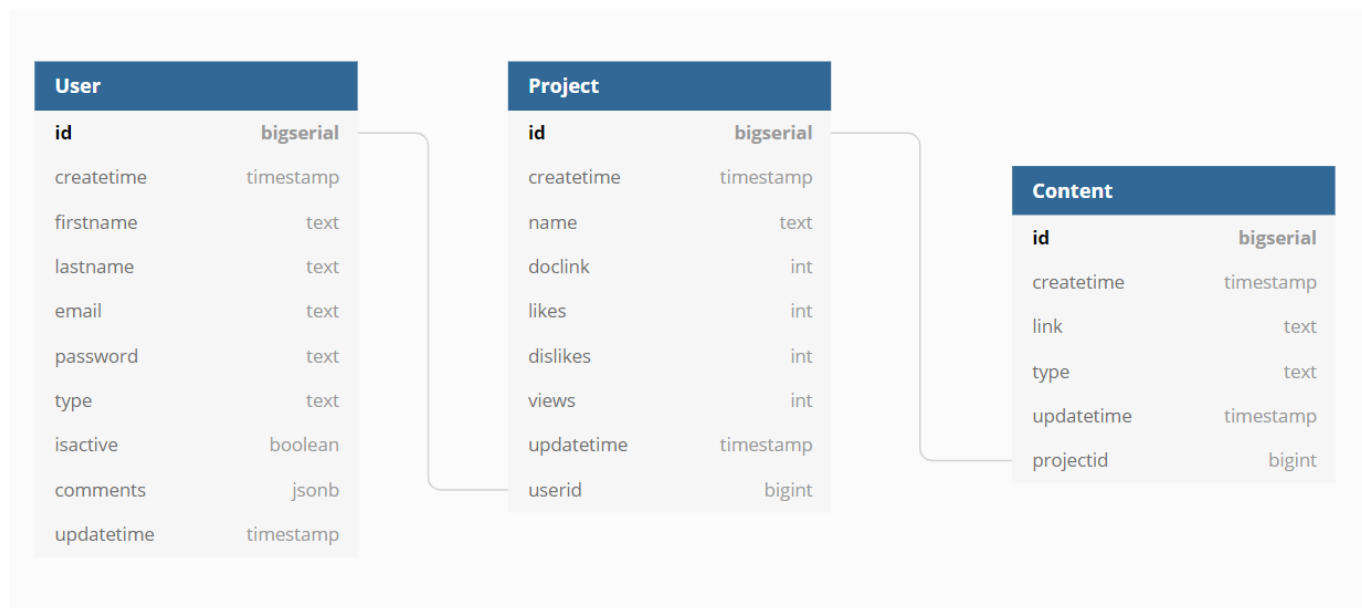
ДОДАТОК А



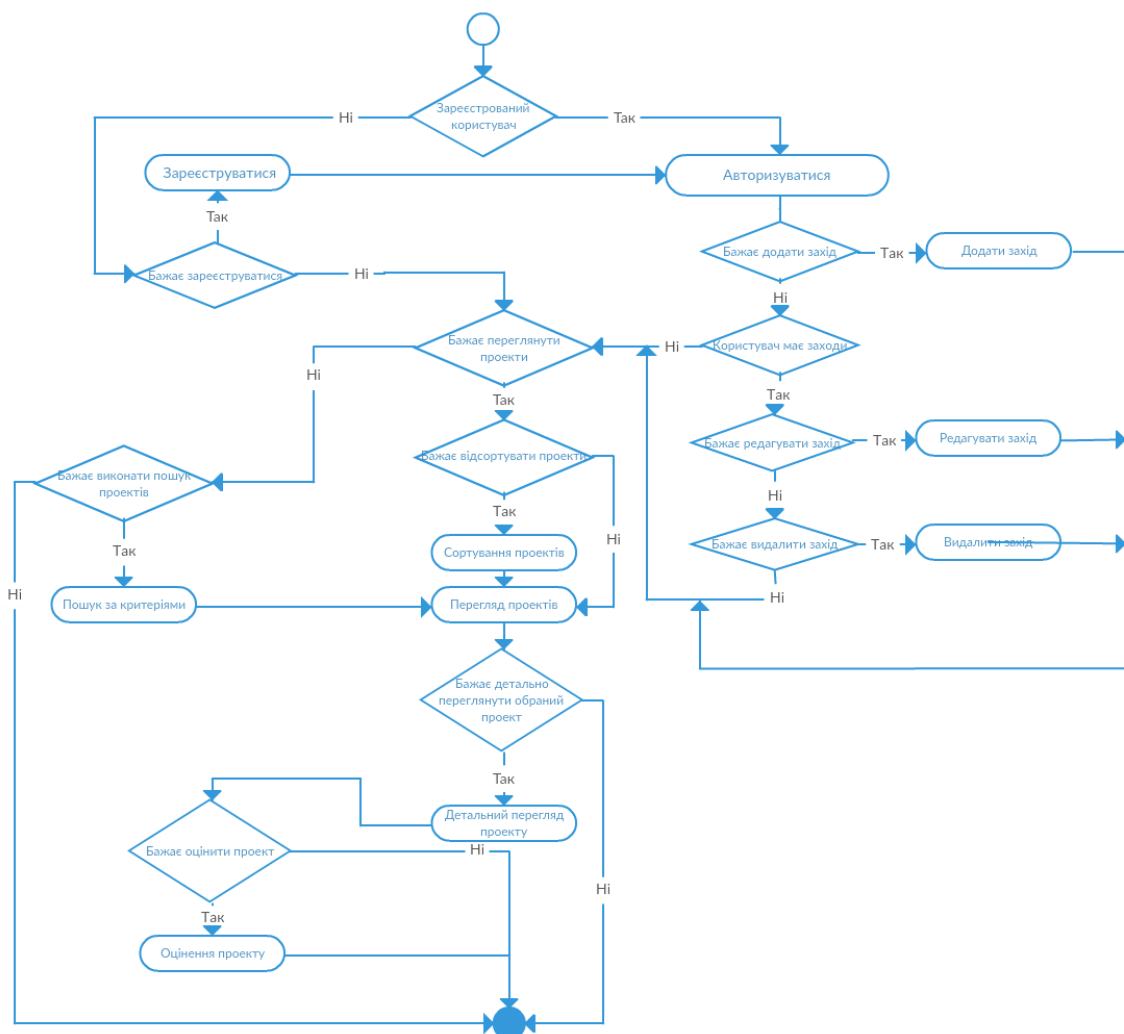
ДОДАТОК Б

User	Project	Content
[id]	[id]	[id]
[firstname]	[name]	[link]
[lastname]	[doclink]	[type]
[email]	[likes]	[projectid]
[password]	[dislikes]	def __init__(self, id, link, type, projectid)
[type]	[views]	@property def link(self)
[comments]	[userid]	@link.setter def link(self)
def __init__(self, id, firstname, lastname, email, password, type)	def __init__(self, id, name, doclink, likes, dislikes, views, userid)	@property def type(self)
@property def firstname(self)	@property def name(self)	@type.setter def type(self)
@firstname.setter def firstname(self)	@name.setter def name(self)	@property def projectid(self)
@property def lastname(self)	@property def doclink(self)	@projectid.setter def projectid(self)
@lastname.setter def lastname(self)	@doclink.setter def doclink(self)	def edit(self)
@property def email(self)	@property def likes(self)	def remove(self)
@email.setter def email(self)	@likes.setter def likes(self)	
@property def password(self)	@property def dislikes(self)	
@password.setter def type(self)	@dislikes.setter def dislikes(self)	
@property def comments(self)	@property def views(self)	
@comments.setter def comments(self)	@views.setter def views(self)	
def deactivate(self)	@property def userid(self)	
	@userid.setter def userid(self)	
	def edit (self)	
	def remove(self)	

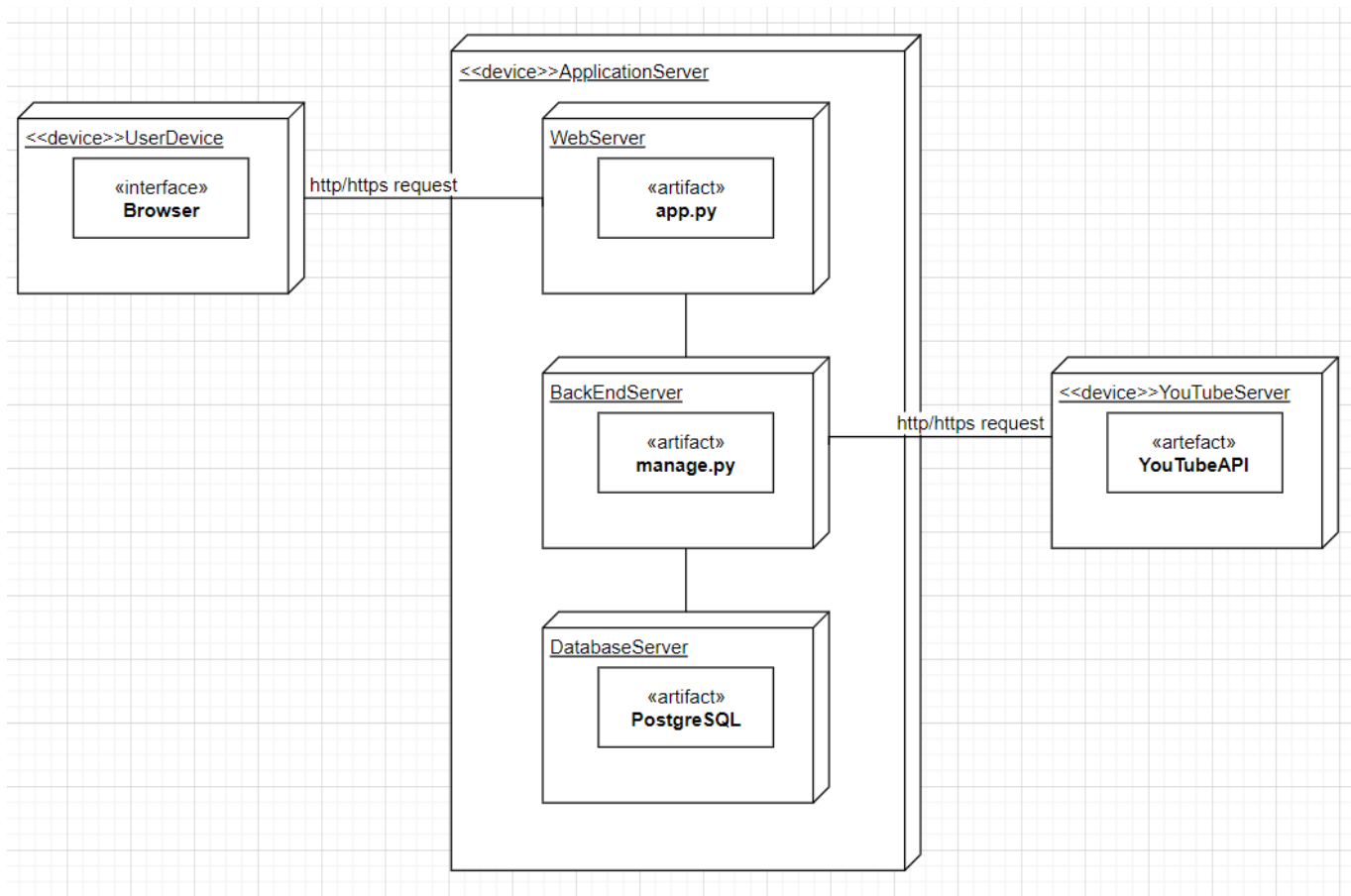
ДОДАТОК В



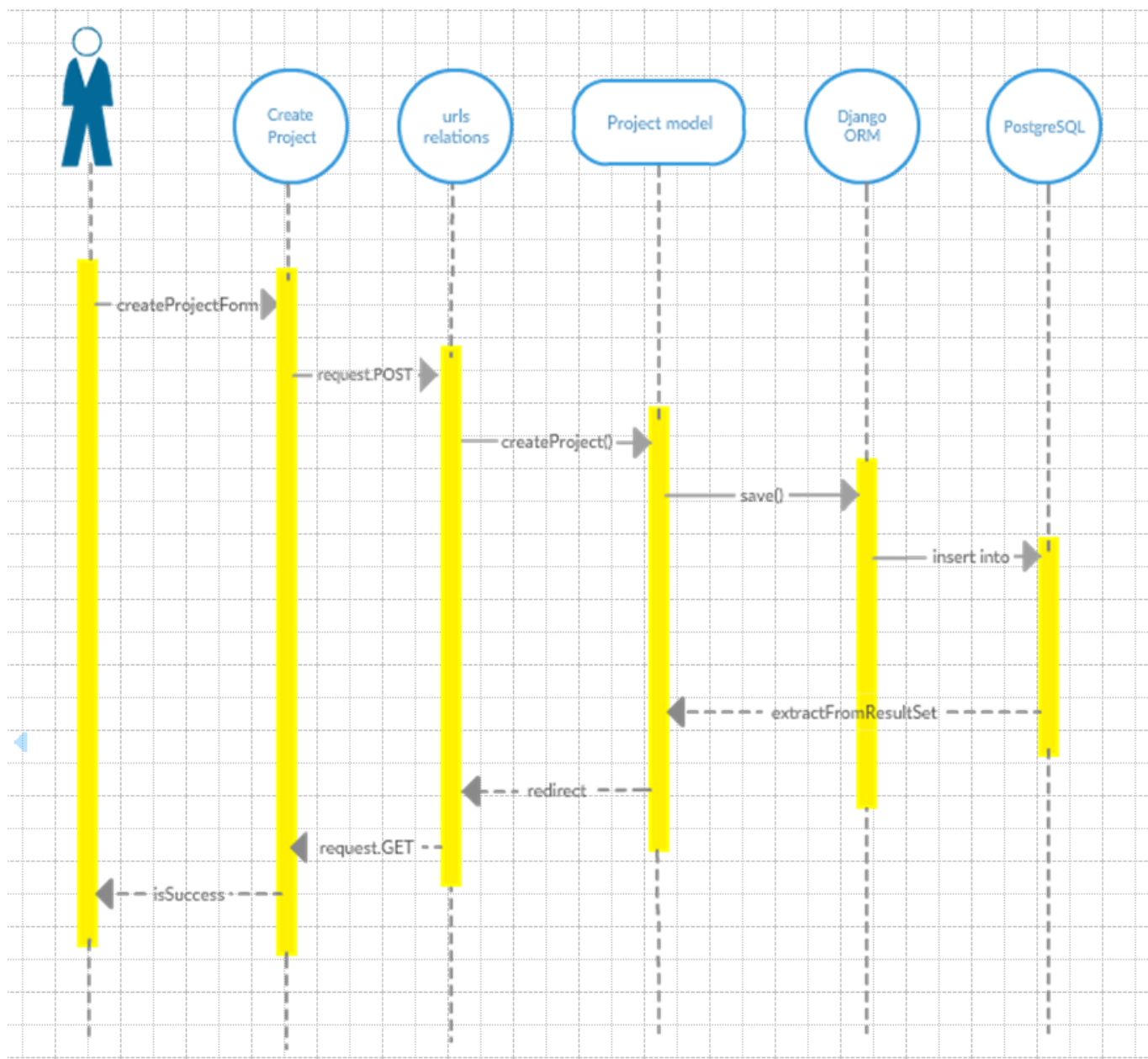
ДОДАТОК Г



ДОДАТОК Д



ДОДАТОК Є



ДОДАТОК Ж



ДОДАТОК 3

